Graduate School of Business

A Framework and Prototype for Intelligent Multiple Objectives Group Decision Support Systems

Jie Lu

This thesis is presented as part of the requirements for the award of the Degree of Doctor of Philosophy of the Curtin University of Technology

March 2000

Abstract

The objectives of this research are threefold: (i) to develop a conceptual framework and a prototype in order to extend the application capability of a category of multiple objective decision support systems (MODSS) techniques; (ii) to explore the combined functionalities of knowledge-based expert systems (ES) and MODSS through embedding an intelligent front-end, and (iii) to develop a new system and process of dealing with multiple objective decision making (MODM) models in a group decision support system (GDSS) framework. Ultimately, a system that integrates MODSS, ES and GDSS is generated, which is then evaluated in a laboratory experimental setup. This integrated system contains a sufficient number of MODM methods to solve MODM problems, provides an ES-based guide to select and use the most suitable MODM method, and has the capability to aggregate individual decision makers' preferences to produce a compromise solution of an MODM problem in different forms and styles of group meetings. The system is supported by a set of group decision making (GDM) methods which combine the preferences of the individual group members and thus increases the confidence of each group member in the compromise solution.

The research is conducted using a multiple-methodologies approach using the system development methodology as the backbone. The conceptual framework of the integrated system is elaborated to integrate multiple system elements into one facility at the application system level based on functional and resource integration. A prototype implements this conceptual framework as an intelligence-based and graphical user interface (GUI)-based MODSS that works in an individual/group environment. Both the conceptual framework and the prototype are called ***Intelligent Multiple Objectives Group Decision Support Systems (IMOGDSS)***.

Initial evaluation of the IMOGDSS is encouraging, which is conducted in the form of testing a number of hypotheses in an experimental setup. This research thus makes

contributions in both theoretical and application domains. **Five major contributions** are listed below:

- It develops a unique conceptual framework of integrating MODSS, ES and GDSS effectively to deal with MODM problem in individual/group decision making under a knowledge-based intelligent architecture.

- It provides a new application of ES, that is, utilising knowledge-based ES to select the most efficient MODM method for each particular decision maker (or decision group) in a particular decision problem.

- The complete method management function of the MODM methodology base guides the decision makers to use the most suitable method to solve their decision making problems, allows them to use multiple methods to resolve complex problems, that could not otherwise be solved with a single MODM, and also allows the group members to get solutions from different methods.

- This study produces an opportunity to select and apply the 'best' aggregation model to aggregate the individual solutions of an MODM problem through integrating various GDM methods in a methodology base.

- This study implements a two-stage configuration of group decision support software that provides a GUI-based hierarchical procedure for solving MODM problems with intelligent guidance in a decision group. The two-stage group decision making procedure is able to help the decision makers to analyse, understand and interact co-operatively in the group decision making process to reach a compromise solution.

Acknowledgement

I would like to express my gratitude to my supervisor Associate Professor Mohammed Quaddus for his supervision and guidance. His helpful suggestions and dedicated assistance have been very invaluable to me in the preparation and completion of this thesis. His encouragement has also led to the publication of five research articles during the period of my study. I would like to express my appreciation to my associate supervisor Robert Williams, Lecture in the School of Information Systems, for his patient advice and generous comments on the system development and programming implementation. I am also grateful to my associate supervisor Dr. Poh Kim Leng, Senior Lecturer in National University of Singapore, for his fruitful suggestions on the expert system design and the combination of the expert system with the multiple criteria decision making.

The work presented in this thesis was done while I held an Overseas Postgraduate Research Scholarship (OPRS) and a Curtin University Postgraduate Scholarship (CUPS) awards. I would like to express my sincere thanks to Australian Government and Curtin University for the financial support.

My heart felt thanks go to my beloved husband, Guangquan Zhang, and my bright daughter, Zoe Zhang, for their company, continuous support and encouragement during the period of my study. I always remember the many weekends they accompanied me in my office. My deep appreciation also goes to my respected father Zesheng Lu and mother Yunxia Zhao, in China, for their selfless love, mental support and company for one year in Perth.

Finally, I would like to express my gratitude to the Graduate School of Business and the School of Information Systems for providing me with the necessary facilities. Acknowledgement is made to all staff at the Schools for their help and encouragement during my study at Curtin University. Thanks also go to my fellow graduate students for their friendship and encouragement during the period of my study.

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Research Questions

Decision Support Systems (DSS) have become increasingly popular as a means of reducing the uncertainty and risk traditionally associated with decision making. Over the past twenty years industry and academia have been working together to research and develop computer based DSS with respect to Multiple Criteria Decision Making (MCDM). At a macro level, MCDM is classified as either discrete or continuous, i.e. Multiple Attribute Decision Making (MADM) and Multiple Objective Decision Making (MODM) problems. Both MODM model based DSS (MODSS) and MADM model based DSS (MADSS) have gained widespread attention and have been classified as two specific types of system within the broad family of DSS. Since MCDM based organisational decisions are primarily taken in a group environment, Group Decision Making (GDM) involving MADM or MODM has also created interests in the literature. To overcome the difficulty of the complex mathematics involved in many DSS models, Expert System (ES) has been, as major technology, designed to combine with DSS. In the mean time Knowledge-Based DSS (KBDSS) or Intelligent DSS (IDSS) have also been widely researched to guide the decision makers systematically towards real world applications.

This thesis focuses on MODM problem domain. Some interesting problems and requirements gaps have been observed in the literature in this field:

- Quite a number of methods exist to deal with multiple objective decision problems, particularly, Multiple Objective Liner Programming (MOLP) methods. These methods are difficult to evaluate and compare since they are based on distinct assumptions about the decision maker's preferences and require different types of preference information. However, it has been observed that some methods are more suitable and appropriate than others in the solution of a particular decision problem for some particular decision makers. This implies that an MODSS with high application capabilities should ***preferably contain a sufficient number of methods*** for the decision makers so that they can choose the most appropriate method for solving their decision problems.

- However, such knowledge for selecting methods is not readily available because general users have a lack of expertise and experience to understand specific features of an available MODM method and have a lack of ability to match the most appropriate MODM techniques with current problem subtasks. Therefore, to utilize the potential of methodology base effectively, an MODSS must ***provide the capability of guiding the decision makers to select the most suitable method*** from the methodology base. A knowledge based intelligent front-end is, therefore, necessary for: (i) identification of the application problem nature and type, (ii) identification of the decision makers' situation, (iii) analysis for current information and the decision makers' desire, and (iv) to determine the best MODM method that matches the needs of the decision maker.

- Since multiple objective decisions are primarily applied in strategic planning of an organization and taken in a decision group environment, Group based MODSS (MOGDSS) has been provided in the literature. MOGDSS should be built to ***provide a group of decision makers with feedback of individual preferences regarding the possible solutions to MODM problems and make an aggregation for all group members' solutions.*** Usually, different group aggregation methods are suitable for different types of group decision making and group meetings. A ***group aggregation methodology base*** will be very useful to: (i) support a wide range of group decision situations, (ii) allow division of MODM tasks, (iii) find the most appropriate method to solve every specific decision problem, and (iv) produce a final compromise solution through multiple methods.

The research reported in this thesis was motivated by the above issues which were not addressed properly in the literature. Specifically, the research centers around the following questions. How is it possible to expand the scope and applications of MODSS to meet these requirements for MODM methodology base, method selection guide, and individual/group decision making? Is it possible to construct a functional integration at the application system level of MODSS, GDSS and IDSS techniques to solve these problems? Going a step further, what is the structure of this integrated system and its components? What should be the balance between MODM, DSS, intelligent technology and GDM in the focus of effort needed to best exploit the opportunities of the integrated system? These questions are considered to generate two formal research questions with three sub-questions as follows:

1. ***What is the effective way to integrate and utilise the combined functionalities of MODM methodology, GDM approach, DSS modelling and Intelligence technology (knowledge-based Expert System) in a conceptual framework?***

   1.1 How is it possible to produce a high quality MODM method management system that is both flexible and responsive to the decision makers and helps the decision makers make better decisions?

1.2 How is it possible to acquire and express the knowledge about MODM method selection, and how to embed a knowledge-component in this integrated system in order to guide the decision makers?

1.3 How is it possible to design the integrated system to help a group aggregate the individual solutions for an MODM application, in order to reach a consensus (a compromise solution) which represents the preferences of whole group, and expresses the most desired solution for the members of the group?

2. ***How can an integrated system be developed effectively consisting of the above technologies to support MODM process in both individual and group environment under an intelligent guidance?***

This study seeks to answer these research questions.

1.2 Objectives

Based on the above research questions, the objectives of this research are (i) ***to develop a conceptual framework and a prototype to extend the application capability of a category of MODSS techniques through providing a complete method management function,*** (ii) ***to explore the combined functionalities of ES and MODSS through embedding an intelligent front-end, and*** (iii) ***to develop a new system and process of dealing with MODM models in a GDSS framework. Ultimately, an integrated system of MODSS, IDSS and GDSS is generated, which contains a sufficient number of MODM methods, and provides the guide to select the most suitable method, and also aggregates individual decision makers' preference to produce a compromise solution supported by a GDM methodology base. Another important objective of this research is to evaluate the integrated system in a laboratory experimented setup.***

The conceptual framework of this integrated system is elaborated to integrate multiple system elements into one facility at the application system level based on functional integration and resource integration. A prototype implements this conceptual framework as an intelligence-based and Graphical User Interface (GUI)-based MODSS that works in an individual and group environment. Both the conceptual framework and prototype are called ***Intelligent Multiple Objectives Group Decision Support Systems (IMOGDSS)***. IMOGDSS, as it is called, uses database, MODM methodology base, knowledge base systems and GDM methodology base, where each decision maker receives a series of guidance during the method selection process based on his/her requirement, and finds the satisfactory solution using the most suitable MODM method. These solutions then are aggregated into a compromise solution supported by the group aggregation methodology base. This compromise solution represents the preferences of the whole group, and expresses the most confident solution for the members of the group.

Integration is the main strategy for implementing the research objectives. There is increasing evidence that integrating the computer-based information systems with one another may enhance the quality and efficiency of many computerised systems in total (Turban 1995). The integration of IMOGDSS is taken in the following two aspects:

Functional integration: integrating the functions of MODSS, IDSS and GDSS so that different support functions are provided as a single integrated system.

Resource integration: integration of database, MODM methodology base, knowledge base and GDM methodology base into an appropriate facility.

Therefore, the integration in IMOGDSS means that the systems are integrated into one facility rather than having separate software, data sources and communications for each independent system. Particularly, functional integration adopted in IMOGDSS implies that different support functions are provided as a single system.

There are two major advantages for IMOGDSS integration:

Enhancements of basic systems. The purpose of such integration is to enhance the capabilities of MODSS. ES is used as intelligent agents to enhance the intelligence of MODSS, and GDM model is used to improve the applications extent and scope of MODSS.

Increasing the capabilities of the applications. In this integration, the systems complement each other. Each system performs subtasks that the system is best at. The integrated framework and prototype will show ES, MODSS and GDSS each performing a different task. Thus this integration can support a wide range of decision situations and reduce the amount of information needed for data inputs.

1.3 Significance

By reviewing the related literature the following observations regarding the IMOGDSS are made:

There is a gap in the research integration of MODSS with GDM

The theoretical framework of MODSS has been researched and a variety of systems prototypes under interactive solution methods have been developed with an assumption of a single user. The literature survey shows that only a very few MODSS are based on a group environment. There is no (as evidenced in the literature) research of MOGDSS that design a group methodology base for providing group aggregation with a flexible system. Therefore, when multiple decision makers are involved in a group environment, the use of different individual preference elicitation technique for a single decision maker and the need for preference aggregation in MOGDSS need further investigation.

There is a genuine lack of MOGDSS in MODM method management

It is found that some methods are more suitable and efficient than others in the solution of a particular decision problem for a particular decision maker. Hence, the proposed MODSS should preferably contain a sufficient number of methods. But currently most MODSS and MOGDSS are specific methodology dependent and they are very specific in nature, i.e. they deal with one specific methodology to analyse the model and obtain a solution. There are no (to the best of researcher's knowledge) MOGDSS which supports the complete method management function with an MODM methodology base, where decision makers are allowed to use different MODM methods to produce a solution fully expressing individual preferences. The integration of MODM methods management function and group decision support framework needs to be explored further.

There is an insufficient application of intelligent technology in MODSS

The literature shows that some researchers have emphasised the need for research regarding the intelligent MODSS, and some studies have explored the possibility of embedding decision guidance within MCDSS (Klein & Methlie 1995). To utilise the potential of a methodology base effectively, MODSS must possess the capability of guiding the user to select and use different MODM methods from a methodology base. An ES as an intelligent front-end is necessary to achieve better guidance. It has been observed that there is no (to the best of research's knowledge) MODSS and MOGDSS using knowledge base techniques to implement such an intelligent support. Therefore, it will be very significant to address such intelligent guidance of selecting the most effective MODM to match actual problem instances in MODSS.

IMOGDSS has not been found in the literature

However, although MODSS, MOGDSS, and IDSS have received strong or weak research attention, the research, development and implementation of IMOGDSS has not been found in the literature.

This research is, therefore, of almost significance as it addresses the above gaps effectively.

1.4 Contributions

The research questions pursued in this study are new, creative and important in the MCDSS and GDSS fields. The research is quite complex as it involves the integration of several kinds of information systems which combines the strong competence of each system and complements the abilities amongst them. This research is conducted by developing the theoretical framework and the associated software implementation. It bridges the gaps between previous and the proposed study.

This study will thus make contributions towards both the theoretical and application aspects. **Five major contributions are listed below**:

- This study develops the conceptual framework of integrating MODSS, ES and GDSS effectively to deal with MODM problems in individual and group decision making under a knowledge-based intelligent architecture. Three dimensions, MODSS, ES, GDSS, are combined to overcome the limitations of each basic system and maximally enhance the competence of the integrated system.

- This study provides a new application aspect of ES, that is, utilising knowledge-based ES to select the most efficient MODM method for each decision maker (or decision group) in a particular decision problem.

- The complete method management function in IMOGDSS guides the decision makers to use the most suitable method to solve their decision making problem, and allow them to use multiple methods to resolve complex problems.

- This study produces and applies the best possible aggregation of individual solutions in MODM through integrating various GDM methods in a methodology base. This work is unique and it not only highlights the different existing GDM approaches to aggregation but also provides new aggregation methods to suit the IMOGDSS.

- This study implements a new configuration of group decision support software that provides a GUI-based hierarchical procedure for solving MODM problems with intelligent guidance in a group environment. The procedure consists of two levels, an individual MODM level and a group aggregation level. This configuration offers a series of advantages.

1.5 A Conceptual Research Framework of IMOGDSS

A great deal of research in the related areas have explored the disciplines and fields that are directly relevant to the concepts of DSS (Eom, Lee & Kim 1993), MODSS (Poh, Quaddus & Chin 1995), KBDSS or IDSS (Klein & Methlie 1995), and GDSS (DeSanctis and Gallupe 1987). MODSS, IDSS and GDSS are three main areas of DSS research family, and these three areas are formed through combining DSS with MODM, AI (or ES) and GDM respectively. Their combinations with each other will produce new concepts of integrated multiple objective GDSS (MOGDSS) (Iz 1992), intelligent MODSS (IMODSS) (Bui & Sivasankaran 1988), and intelligent GDSS (IGDSS) (Limayem & Chelbi 1997), even though these terms were not directly used. IMOGDSS is located at the intersection of MODSS, IDSS and GDSS.

The conceptual research framework presented here is used to gain an understanding of the positioning of IMOGDSS in the hierarchy of DSS and a clearer definition of its purpose and design requirements. Figure 1-1 shows the framework. With the variety of AI, MODM and GDM techniques, IMOGDSS of various features are obtainable.

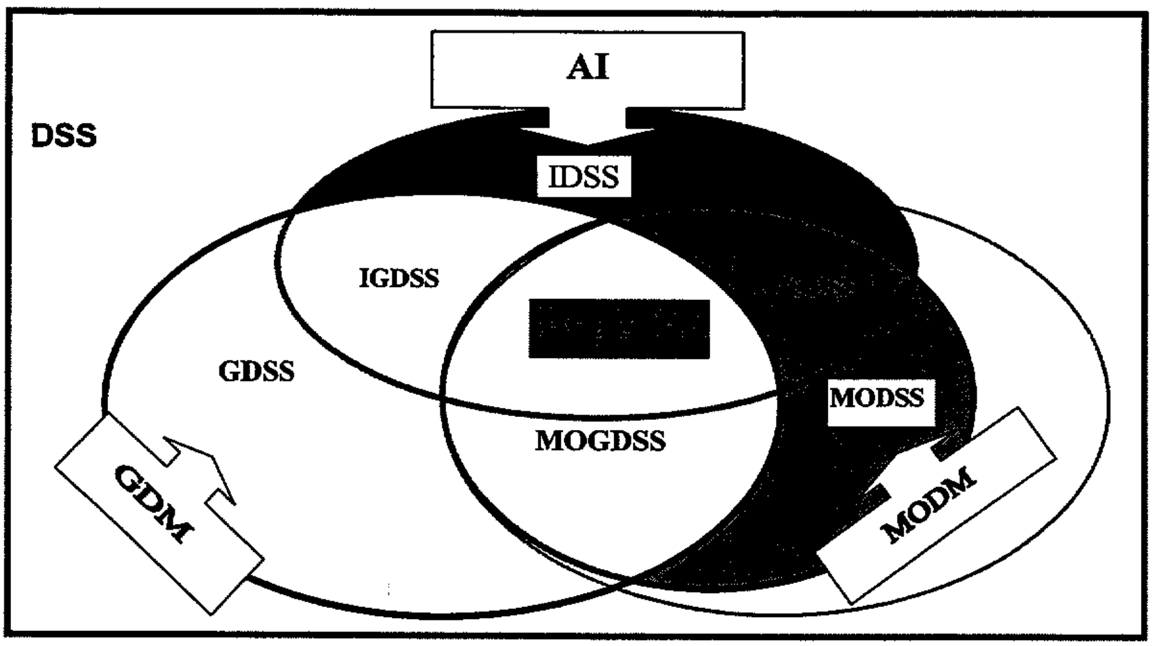


Figure 1-1 A conceptual research framework

1.6 Scientific Background

To understand how IMOGDSS is researched, a body of knowledge of this research is described. The scientific background that has led to the definitions of the three system concepts, IDSS, MODSS and GDSS are shown, diagrammatically, in Figure 1-2. Some key concepts that characterise each major scientific development are also shown.

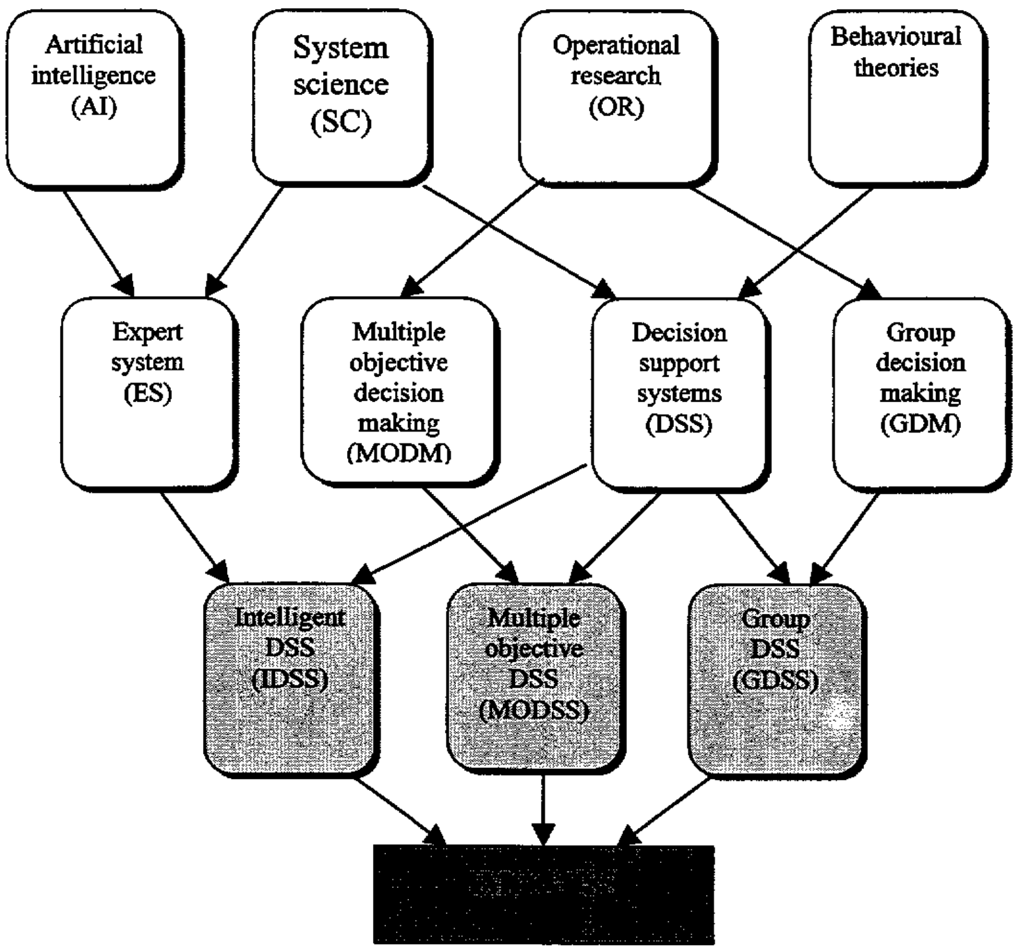


Figure 1-2 The scientific background

1.7 Contents of This Thesis

This thesis comprises 10 chapters (Figure 1-3).

Chapter 1 presents an overview of this research.

Chapter 2 reviews the related research areas, particularly addresses the research background with regard to MODM, MODSS, IDSS, GDSS and MOGDSS.

Chapter 3 discusses research methodologies and research design of this study.

Chapter 4 proposes a conceptual framework of IMOGDSS to deal with MODM application for individual decision makers or decision group environment under an intelligent guidance, which is called IMOGDSS.

Chapter 5 presents the design and implementation of IMOGDSS prototype.

Chapter 6 explores the possibility and advantages of embedding an intelligent front end within MOGDSS, and outlines a specific framework for developing such a knowledge-based intelligent guide subsystem. This intelligent guide subsystem is used to recommend the most appropriate method to decision makers in solving various MODM problems based on their situation.

Chapter 7 describes MODM method management functions with a methodology base that allows flexible use of seven popular MODM methods to solve various multiple objective linear decision problems.

Chapter 8 proposes a group aggregation subsystem embedded in IMOGDSS. The group subsystem is designed to include a GDM methodology base to aggregate a group of MODM solutions as alternatives that involve multiple conflicting objectives and are provided by multiple decision makers with different business functions and preferences. This group subsystem provides a new configuration of group decision support software.

Chapter 9 discusses the testing and evaluation of IMOGDSS framework and prototype.
Chapter 10 recapitulates the whole thesis and future research.

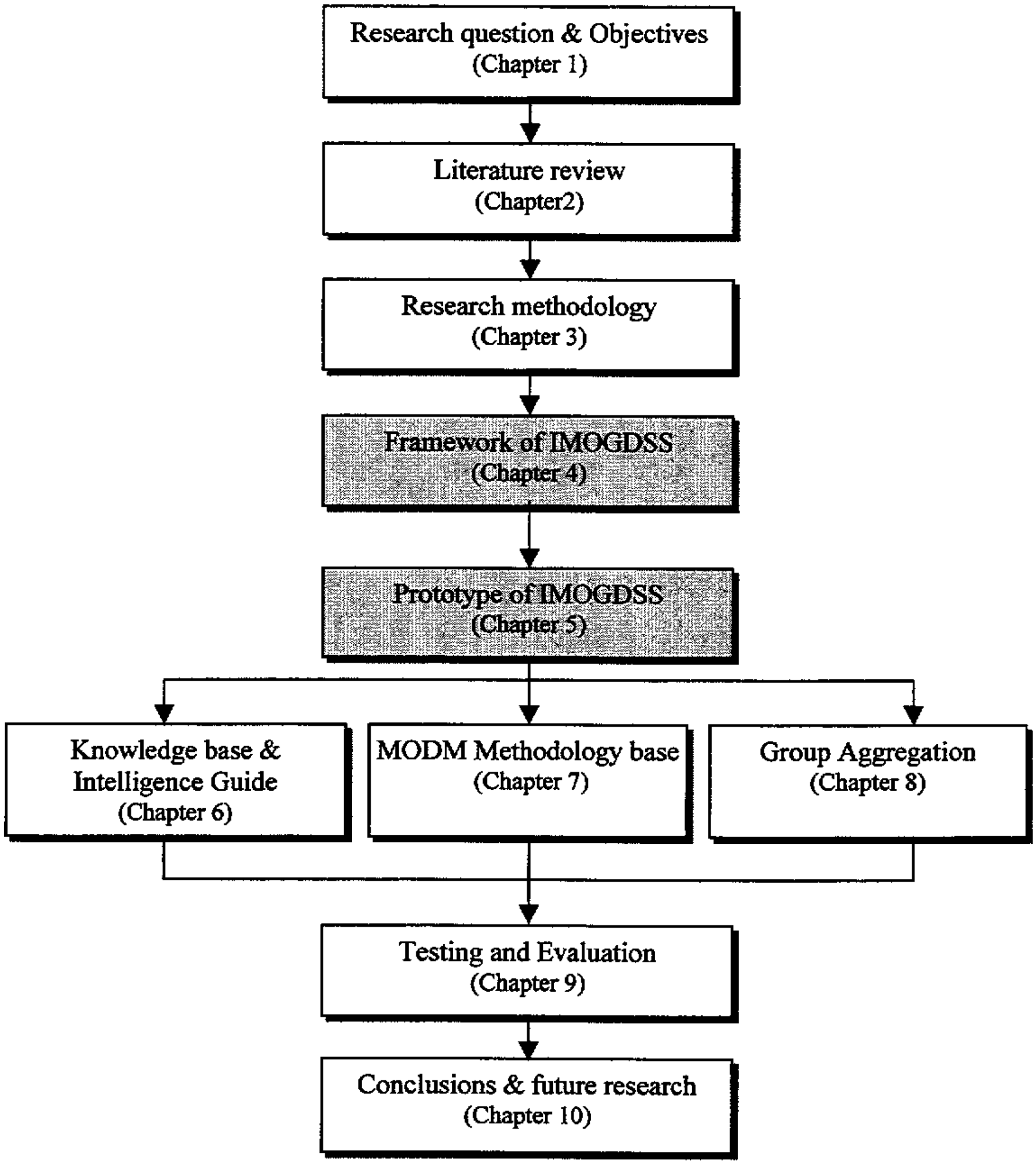


Figure 1-3 Organization of the thesis

1.8 Summary

This chapter presents an overview of this research entitled ***"framework and prototype of intelligent multiple objectives group decision support system"***. The research questions and objectives are addressed. Then the significance and contributions of the study are pointed out. The research contents are then outlined. The chapter aims at acquainting the readers with the holistic picture before elaborating on the research theme in the subsequent chapters.

Chapter 2

LITERATURE REVIEW[1]

2.1 Introduction

A review of the literature is essential in order to ascertain the research work that has been carried out in the field of the present study. It highlights the areas in which detailed analysis has already been done and also reveals the topics in which further research can be fruitfully made.

The Intelligent Multiple Objectives Group Decision Support System (IMOGDSS) is an intersection of the Multiple Objectives Decision Support System (MODSS), Intelligent Decision Support System (IDSS) and the Group Decision Support System (GDSS). The MODSS, IDSS and GDSS, as three important branches of DSS, are contributed to by Multiple Objective Decision Making (MODM), Artificial Intelligence (AI) and Group Decision Making (GDM) respectively. Therefore, the literature review includes these

[1] This chapter is based on the following paper:
Lu, J., M.A. Quaddus and R. Williams, "Intelligent multiple objectives group decision support systems: a review, prototype and an application", *Proceedings of Acer National Business Education and Research Conference,* 14-15 October, 1999.

related research areas. The literature of MODM is first reviewed in Section 2.2. Decision making, in general, and Multiple Criteria Decision Making (MCDM) are discussed in this section. DSS is reviewed in Section 2.3. The literature shows that over the past twenty years industry and academics have been working together to research and develop computer based DSS to increase decision making quality, and fruitful results have been achieved. MODM model-based DSS, (MODSS), has gained widespread attention and has been classified as a specific type of system within the broad family of DSS. MODSS is reviewed in Section 2.4. To overcome the difficulty of the complex mathematics involved in many MODSS models, the Knowledge-Based DSS (KBDSS) or Intelligent DSS (IDSS) have been widely researched to guide decision makers systematically towards real world applications. Section 2.5 investigates this focus. Since organizational decisions are primarily taken in a group environment, DSS involving Group Decision Making (GDM) has been researched and is called Group Decision Support System (GDSS). GDSS involving MCDM (MOGDSS) has also created interest in the literature. The GDSS and MOGDSS are discussed in Section 2.6.

2.2 Multiple Objective Decision Making

This section produces a detail account of decision making in general and multiple objective decision making in particular.

2.2.1 Decision Making

Decision making, as an activity, is undertaken by individuals or groups in every walks of life. This subsection sets the sieve for decision making process.

2.2.1.1 Decision Making Process and Decision Makers

In this study we shall use the term *decision making* to include all stages of problem solving such as: recognizing situations that call for actions, formulating problems, designing actions and setting goals, as well as final evaluation and choice. We call these tasks the *decision making process*.

The successful manager is able to choose the right actions at the right time. Thus, our usual image of a *decision maker* (DM) is of a person who makes the right choices. However, a choice is just the final result of a complex process of exploration, analysis and evaluation.

The decision making process can be described by models which form a continuum from the rational models at one end, to irrational models at the other end (Klein & Methlie 1995). The difference is very much determined by the way that the decision maker or the decision making body (in the case of a group) is treated. Rational models regard the decision making body as homogeneous. That is, a person or a group of persons with consistent and non-conflicting objectives. The irrational or anarchistic models, on the other hand, regard the decision making body as a heterogeneous group of people. That is, people have changing and conflicting preferences (Klein & Methlie 1995).

2.2.1.2 Ill-Structured Decision Problems

Types of informational reports needed by three management levels: strategic (top management), tactical (middle management) and operational (lower management), for planning, control and other organization activities are discussed by many researchers (Thierauf 1982). With the level of management changing from the operational management to tactical level, then to strategic level, the decision making problem becomes semi-structured to unstructured. These two kinds of decision making problems are called ill-structured decision making problems (Sprague & Hugh 1995). Klein (Klein & Methlie 1995) provides six characteristics of ill-structured decision making problems as follows:

- The preferences, judgments, intuition and experience of the decision makers are essential.
- The search for a solution implies a mixture of:
    - search for information;
    - formalization, or problem definition and structuring (system modelling);
    - computation;
    - data manipulation.
- The sequence of the above operations is not known in advance since:
    - it can be a function of data;
    - it can be modified, given partial results;
    - it can be a function of the user preferences.
- Criteria for the decision are numerous, in conflict, and highly dependent on the perception of the user (user modelling).
- The solution must be achieved in limited time.
- The problem evolves rapidly.

As can be noticed from the above description, one of the key characteristics of situations where decision support tools are useful is where modelling of the decision problem (and users) is needed. That is, decision makers need tools, models and technology to assist them in dealing with ill-structured problems (Sprague & Hugh 1995). Therefore the tools, models and technology are a prerequisite. Very often the decision problems have to be solved in limited time. In such a situation it is crucial to be able to implement a DSS application which provides a first satisfactory solution and which can be improved as time permits (Klein & Methlie 1995). Obviously, it is necessary to provide the tools, models and technology to assist the decision makers in dealing with ill-structured decision making in an allowed time.

2.2.2 Multiple Criteria Decision Making (MCDM)

In most real world contexts, decisions must be made in the presence of multiple, conflicting, incommensurate criteria. Particularly, many decision problems at tactical and strategic levels, such as strategic planning problems, have to consider explicitly the models that involve multiple conflicting objectives/goals (or criteria) (Massam 1988; Rietveld 1980). The problems are then analysed in detail and a 'choice' of the proposed solutions is made for implementation. This kind of decision problem is called Multiple Criteria Decision Making (MCDM). MCDM has been characterized in several ways, reflecting the growth of the discipline, and refinement of concepts employed therein. For the purposes of this thesis, MCDM will constitute both descriptive and prescriptive models of decision making involving multiple attributes, objectives, and goals (Nazareth 1993). As one of contributing disciplines for operations research, MCDM has been one of the fastest growing areas in operations research during the last two decades (Korhonen, Lewandowki & Wallenius 1991; Massam 1988; Rietveld 1980; Shin & Ravindran 1991; Steuer 1986). A major reason behind the recent development in this area is due to the large number of criteria that today's decision makers are expected to incorporate in their actions. Their multiple and incommensurate concerns often include economic, political, environmental, and social criteria which necessitate compromise among the conflicting criteria and objectives (Iz & Jelassi 1990).

Balestra and Tsoukias (1990) presented a descriptive model of the decision aid process based on these assumptions, considering that a multiple criteria approach could be the more appropriate 'paradigm' (Roy, 1987b) for facing the complexity of different decision situations. In this model five activities of decision aid are divided as:

- identify, define and modify the decision situation;
- acquire information significant to the decision situation, modify the communication system in the decision situation;
- build a model of the decision situation, shared and accepted by the client, correctly defined by a formal methodology used by the analyst;
- use coherently an appropriate method of decision aid;
- acquire and construct useful and significant data for the method adopted.

Thus it is possible to identify a model under which such activities are performed, that there exists an appropriate formalism for this model and that it is possible to implement it in a computer program. So, a DSS using multiple criteria concepts and methodology can be developed reproducing some activities of the decision aid process represented and implemented in appropriate formalisms and languages.

Different frameworks to classify MCDM are available in the literature, for example see Korhonen et al. (1992), Hwang and Masud (1979), Steuer (1986), Shin and Ravindran (1991), Quaddus and Holzman (1986), among others. The variety of MCDM techniques proposed in the literature can be divided into four categories:

- procedures based on multiple objective mathematical programming, i.e. MODM;
- procedures based on multiple attribute utility theory, i.e. MADM;
- procedures based on outranking relations;
- procedures based on analytical hierarchy process.

In another classification, MODM is also called a discrete MCDM and MADM can be classified as a continuous MCDM (Korhonen, Moskowitz & Wallenius 1992).

2.2.3 Multiple Attribute Decision Making (MADM)

MADM draws heavily from rational decision analysis, including such areas as multiple attribute utility theory (Keeney & Raiffa 1976), game theory and some aspects of behavioural decision theory (Nazareth 1993). Additionally, it may employ economic principles, as in marginal rates of substitution in hierarchical tradeoffs, and also statistical clustering principles, in the case of multiple dimensional scaling. MADM problems are rather diverse in nature as they may range from the selection of an apartment by an individual to the allocation of resources within a company. However, almost all MADM problems have the following common characteristics (Poh 1998).

- they involve the selection of the best alternative from among a set of predetermined alternatives;
- they involve the use of multiple attributes in the evaluation process;
- they are required to resolve conflicts and value trade-off among the attributes;
- they involve the use of incommensurable units that cannot be compared directly. Most MADM methods select the best alternative based on inter and intra-attribute comparisons. These comparisons may involve explicit or implicit trade-off.

Mathematically, the MADM problems can be represented as follows:

$$\left.\begin{array}{l} Select\!: alternative\ A_1, alternative\ A_2, \ldots\ldots alternative\ A_m \\ Subject\ to\!: attribute\ C_1, attribute\ C_2, \ldots\ldots attribute\ C_n \end{array}\right\} \quad (2.1)$$

where $A=(A_1,\ldots\ldots,A_m)$, denotes m distinct alternatives, $C=(C_1,\ldots\ldots,C_n)$, represents n attributes. The *select* is normally based on maximizing a multiple attribute value (or utility) function elicited from the stakeholders. In the literature these types of decision making are also known as discrete MCDM (Belton 1989). The classic book by Keeney and Raiffa (1976) covers MADM problems in greater detail. There are different schools of thought to develop the value function needed to analyse problem (2.1) above. On the one hand a rigorous approach is suggested to test various conditions (i.e. preference independence) and develop a comprehensive multiple attribute value function (Keeney & Raiffa 1976). On the other hand a simple linear weighted sum approach is suggested to develop the multiple attribute value function (Edwards 1977; Belton and Vickers 1989). The Key to this simple approach is the interactive sensitivity analysis to clarify uncertain and fuzzy ideas until a requisite model is developed. From the applied point of view this simple approach deserves attention. Many variations of this simple approach are available in the literature that are being applied. A survey by Hwang and Yoon (1981) reported at least a few dozen such methods prior to the beginning of the 1980s.

2.2.4 Multiple Objective Decision Making (MODM)

MODM is widely known as the continuous type MCDM. The problems in this category are characterized by a large number of implicit alternatives i.e. alternatives are hidden in the functional form of the constraints. The objectives are also defined in clear functional forms. The MODM has its roots in vector optimisation. Subsequent development has seen additions from goal programming (Ignizio 1976), to worth tradeoffs and refinements in mathematical programming with multiple objectives (Zionts & Wallenius 1976). The development in the MODM area has been attributed to the decision activities where there are a large number of conflicting objectives that the decision makers are expected to incorporate in their various decision making actions (Iz & Jelassi 1990). MODM problems have the following common characteristics (Hwang & Masud 1979; Shin & Ravindran 1991; Quaddus & Siddique 1996a):

- they involve the selection of the best satisfactory solution from among a set of efficient solutions;

- they involve the use of multiple constraints in the maximisation/minimisation process;
- they are required to resolve conflicts and value trade-off among the objectives; and
- they involve the use of interactions from decision makers to arrive at a satisfactory solution.

An MODM model considers a vector of decision variables, and develops the objective functions and constraints. The decision makers attempt to maximize (or minimize) the objective functions. Since this problem has rarely a unique solution, the decision maker is expected to choose a solution from among the set of efficient solutions. Generally, the MODM problem can be formulated as follows:

$$\left.\begin{array}{ll} \textit{Maximize} & \{ f_i(\overline{x}),\ i = 1, \ldots\ldots, n \} \\ \textit{Subject} & \textit{to} : g_j(\overline{x}) \leq b_j,\ j = 1, \ldots., m \end{array}\right\} \qquad (2.2)$$

Where $\overline{x} = (x_1, x_2, \ldots, x_k)$ denotes k-dimensional vector of decision variables, $f_i(\overline{x})$, $i=1,2,..,n$ represent n conflicting objective functions, and $g_j(\overline{x}) \leq b_j$, $j=1,2,..,m$ represent m constraints.

Theories and methodologies on MODM are widely available in the literature, for example, see Hwang & Masud (1979), Steuer (1986), among many others. All these methodologies need interactions of some kind from the decision makers to arrive at a compromise (and satisfactory) solution. Some interactions are performed before the solution process even starts. In this type, explicit preference functions of the decision makers are needed. In others, the preference information of the decision makers are needed during the solution process. In this type, the decision makers are required to provide on line preference information, but no explicit preference function is developed. The third type of approach requires preference information after a large set of candidate solutions has been generated. In this the planners are simply required to choose the most satisfactory solution from the set (Quaddus 1997).

The multiple objective mathematical programming procedures can be further classified according to the assumptions made on the variables (continuous or integer), on the type of functions by which the objective and constraints are defined (linear, non-linear, convex, non-convex, differentable, etc.). For every case, there is a set of methodologies that can be used to solve it. The recent technological advances in computer software offer a lot of potential for future applications (Iz & Krajewski 1992). Some recent research in MODM concentrated on discussing and producing new algorithms and techniques to solve unconstrained MODM problems. For example, Binh (1999) provided a multiple objective evolutionary algorithm. Czyzak and Jaszkiewicz (1998) developed a simulated annealing technique. These new algorithms and techniques are still in the development process. However they already proved themselves as successful techniques for solving unconstrained MODM problems.

The MADM and MODM approaches are quite different in the type of problems that they deal with. MODM method is suitable for more structured multiple criteria problems. It is more complicated and has a higher dependence on models and algorithms. It therefore needs more expertise to develop the structure appropriate for MODM problems and has a stronger requirement for tools' support. On the other hand, structuring a problem for MADM method is simple and more natural.

2.3 Decision Support Systems

2.3.1 Definition of Decision Support Systems (DSS)

Since the term DSS was coined in the early 1970s, a growing number of studies in the area over the past two and a half decades have been reported (Eom 1998). The topic of DSS has stimulated great interest in its research and application (Bonczek et al. 1981; Keen 1987; Sprague 1987; Sol 1985; Sol 1987; Radermacher 1994). Right from the start of the DSS movement, and even now, there has been no established definition of DSS. It is not at all unusual for attendees at conferences on the topic to say that they came to find out what DSS is, since researchers are talking about it a lot, and at the end it is still unclear to them (Keen 1987). Actually, DSS have been defined in several ways. The basis for definitions include objectives, functions, components, usage patterns, capabilities, and development processes (Sprague 1980; Radermacher 1994; Sprague & Watson 1989; Turban 1996; Keen 1987; Hudson, Cohen & Deedwania 1994; Bonczek, Holsapple & Whinston 1981). The definitions of DSS emphasise various aspects of the DSS concept, and are targeted at different audiences. However, the definition of DSS that we apply in this thesis was articulated in a "framework" paper on DSS (Sprague 1980). This definition of DSS is described as:

- computer-based systems;
- that help decision makers;
- confront ill-structured problems;
- through direct interaction;
- with data and analysis models.

Each part of this definition has a key concept that contributes to the unique character of DSS. DSS are computer- based systems which add to the many other approaches and tools that decision makers can use to assist in decision-making, and there may be no optimum solution because these are ill-structured problems and situations, so the decision must evolve through the interaction of decision makers with resources such as data and analysis models.

A further discussion for the balance between D, S, and S of DSS has been made by Keen (Keen 1987). 'Decision' relates to the non-technical, functional and analytic aspects of DSS and to criteria for selecting applications. 'Support' focuses on implementation and understanding of the way real people operate and how to help them. 'System' directly emphasizes skills of design and development and of technology. If we do not focus much more specifically on the 'decision' component, the drift towards commodity systems building instead of value-added expertise will continue. If we do not look for definitions for understanding and action that open up extended 'support', the organizational role for DSS will narrow not broaden. If we neglect the many new developments in technology, we will limit the scope for DSS in terms among 'decision', 'support' and 'system'. Therefore, the relevant research problems have been identified in this area such as, different approaches to the building of DSS, different methodologies used by DSS and different tools used to formulate the decision support and to implement the DSS (Te'eni & Ginzberg 1991; Sprague 1987; Angehrn & Jelassi 1994; Kasper 1996; Rudowski, East & Gardner 1996; Sequeira, Willers & Olson 1996; Khoong 1995; Xu 1994).

2.3.2 Features of DSS

The notion of a DSS to aid managers in ill-structured decision situations has been formulated for more than two decades. The idea that a DSS is intended to support, rather than replace, managerial decision making in ill-structured decisions with a view to improving decision making effectiveness, rather than efficiency, still forms the basic premise for most DSS (Nazareth 1993). A variety of frameworks and paradigms have been proposed to ground and construct DSS (Bonczek, Holsapple & Whinston 1981). The literature on DSS has always had an emphasis on increased effectiveness of decision making, that is, an increase in quality of the decision, as the main benefit of a DSS (Eom 1998; Radermacher 1994). Also, the literature indicates that it is possible to use computers to assist in problem solving performance and decision making and this support can lead to better decisions (Angehrn & Jelassi 1994; Chaudhry, Salchenberger & Beheshtian 1996). Some evaluation researches propose effects of DSS on decision outcomes development. These studies evaluate the improvements in decision quality typically associated with DSS which are due primarily to 'development' or 'reliance' effects; some researches also examined how the introduction of DSS contributes to decision quality after controlling for task familiarity (Barr & Sharda 1997). A good DSS environment improves the decision making process, by speeding up the learning process of the decision makers and providing reliable methods (Nazareth 1993; Bonczek 1981; Klein 1985; Sprague 1987).

The main features of a DSS are summarized as:

- interactive and extremely user friendly;
- highly flexible in carrying out a decision support task; and
- dedicated (or can be made dedicated) to support a specific decision-making activity by using its methodology base, data base, knowledge base (rule base) and other facilities.

As reported by Sprague and Watson (Sprague & Watson 1989), DSS is used to support decision making activities in organizations for semi-structured to unstructured problems, i.e. in situations where human intervention is necessary. This human intervention may come in many different ways:

- during the problem structuring;
- during the *solution process*; and
- during the phase of desirable choice making.

It is observed that although the domain of DSS is semi-structured to unstructured problems, some form of structure is provided to the problems during the problem and model structuring process. The extent and type of this structure is dependent on the model being used for information generation.

2.3.3 Principles of DSS

During early development of the DSS field several principles evolved. Eventually, these principles became a widely accepted 'structural theory' or framework (Sprague 1987). The three most important of these principles are summarized below as DDM paradigm, DSS technology levels, and DSS development environment (Sprague & Hugh 1995).

2.3.3.1 The DDM Paradigm

The technology for DSS must consist of three sets of capabilities in the areas of dialog (D), data (D) and modelling (M), what Sprague and Carlson called the DDM paradigm (Sprague 1987). They make the point that a good DSS should have a balance among the three capabilities. It should be easy to use to allow non-technical decision makers to interact fully with the system; it should have access to a wide variety of data adopted in the DSS; and it should provide analysis and modelling in a variety of ways where they are strong in only one area and weak in the others (Angehrn & Jelassi 1994). Figure 2-1 shows the relationship between these components and the three components of the dialog function. The data base management system (DBMS) and the model base management system (MBMS) contain the necessary functions to manage the database and model base, respectively. The dialog generation and management system (DGMS) manages the interface between the user and the rest of the system.

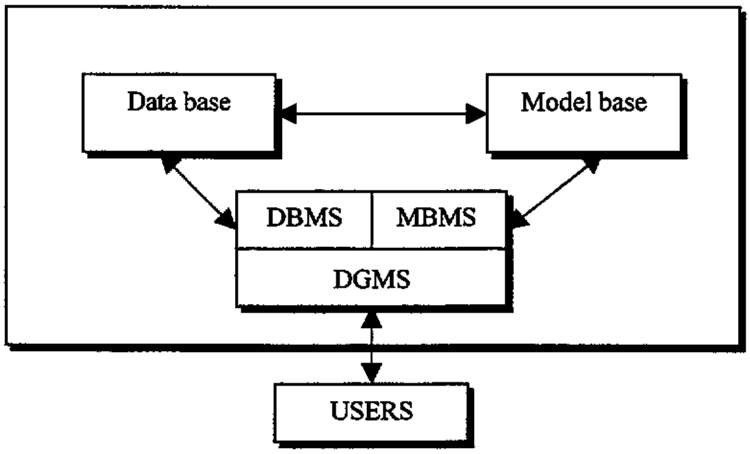


Figure 2-1 The components of DSS

2.3.3.2 DSS Technology Levels and Environment

Three levels of technology are useful in developing DSS. This concept illustrates how to configure and develop a variety of specific DSS to aid decision makers (Figure 2-2). The system that actually accomplishes the decision support work is called the specific DSS (such as multiple objective DSS), shown as the top level of the diagram. It is the hardware/software that allow a specific decision maker(s) to deal with a set of related problems. It can be developed either directly from tools or by adapting the DSS generator. The DSS generator development approach can be represented by the interactive cycling between the DSS generator and the specific DSS (see (3) in Figure 2-2). The second level of technology is called the DSS generator. A DSS generator is a package of related hardware and software that provides a set of capabilities to quickly and easily build a specific DSS (Bonczek 1980). Such generators often integrate with synergy the computer technologies and solving methodologies (model solver, query processor, hypertext engine, report generator, etc.) required. Examples of such generators include IFPS, Prefcalc and Expert Choice (Sprague & Hugh 1995). The third level of technology is DSS tool that are hardware and software elements built by a toolsmith to facilitate the development of either a specific DSS or a DSS generator (Sprague & Hugh 1995). This can be seen respectively in (1) and (2) in Figure 2-2. Examples of such DSS tools include procedural programming languages, graphics and other dialog-handling software.

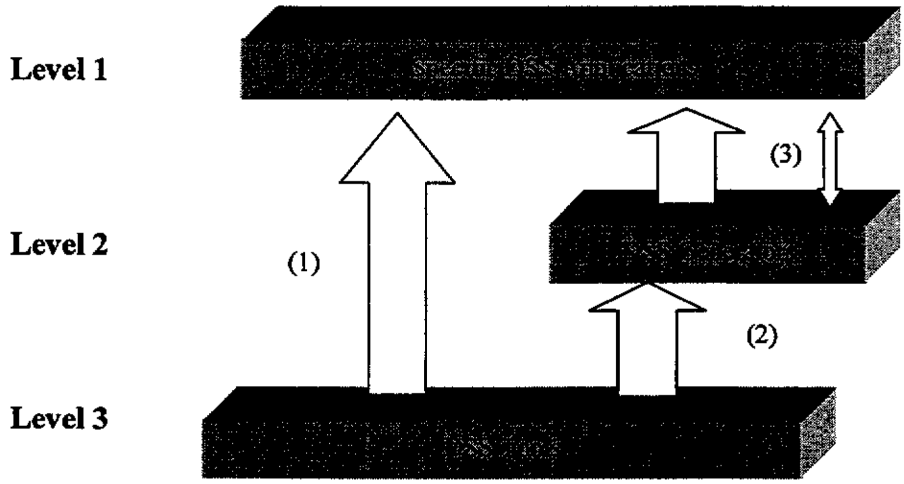


Figure 2-2 Three levels of DSS technology

The nature of DSS requires a different design and development technique from traditional batch or on-line systems (Sprague & Hugh 1995; Radermacher 1994; Pitty & Reeves 1995). Effective development of DSS requires an environment which includes a group of people with interaction roles, a set of hardware and software technology, a set of data sources, and a set of analysis models (Sprague & Hugh 1995; Radermacher 1994; Pitty & Reeves 1995; Xu 1994; Watson & Buede 1987). DSS must be developed to permit the change quickly and easily.

2.3.4 Major Areas of DSS Research and Practice

The role and benefits of DSS are widely recognized today around the world. A 1987 survey of management information system (MIS) researchers reported that DSS was one of the five most popular research themes from 1977 to 1985 (Farhoomand 1987).

Another study reported that nearly one third of the researchers in MIS were doing DSS research (Teng & Galetta 1990). A more recent major study completed reported a view of the intellectual structure of the DSS field (Eom, Lee & Kim 1993). Drawing from various reported success and failure stories of DSS deployment, Khoong (1995) developed an extended research agenda for driving DSS research and development into the next century. The guiding philosophy in the agenda is the concept of anywhere, anytime, anybody decision-making.

Throughout the 80s, there were strong advances in several areas of technology that combined to affect the field of DSS. Each of these trends had a significant effect on the growth and development of DSS. The relationships between the decision support system subspecialties and reference disciplines are investigated by Eom (1998), Angehrn and Jelassi (1994), among others. These studies systematically identified the DSS reference disciplines and trace how concepts and findings by researchers in the contributing disciplines have been picked up by DSS researchers to be applied, extended, and refined in the development of DSS research subspecialties. Six major areas of DSS research are identified systematically (Eom 1998):

- Group DSS (GDSS);
- Foundation;
- User Interfaces;
- Model Management;
- Multiple Criteria DSS (MCDSS);
- Implementation.

DSS has obtained a set of significant contributions from relevant practice fields and research disciplines for its growth. From the practitioners' perspective, the fundamental concepts of DSS (scope and objectives, design and implementation, applications) have been gradually introduced by many relevant issues and their integration. Such issues included focusing on decision making processes, integrating DSS into the organizational context, and taking advantage of new teamwork (group)-oriented concepts and technologies (Angehrn & Jelassi 1994). From the researchers' perspective, the DSS is linked with disciplines such as decision sciences, organizational behaviour, and cognitive sciences and has contributions by disciplines such as organization science, management science/operations research, artificial intelligence and MCDM. A major study completed by Eom (1993) developed a view of the intellectual structure of the DSS field and proposed six contributing disciplines:

- MCDM;
- Cognitive Science;
- Artificial Intelligence (AI);
- Organization Science;
- Group Decision Making (GDM);
- System Science.

Figure 2-3 shows the set of disciplines that are directly and potentially relevant to the concept and design of DSS.

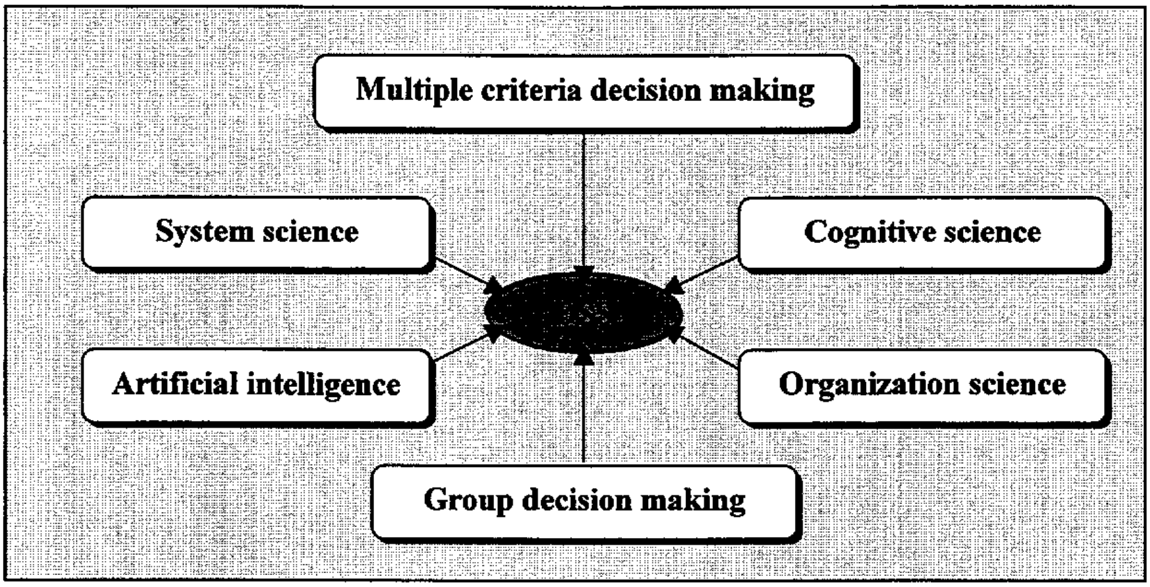


Figure 2-3 DSS and the relevant disciplines

It has been found from the literature that the MCDSS, IDSS and GDSS are three of the main directions in which both DSS research and practice might evolve, and MODM, AI and GDM are three of the main contributing disciplines.

2.3.5 Design and Implementation of DSS

The design, implementation and system development methodology of DSS have been explored by many researchers (Buede 1992; Kasper 1996; Moreno 1994). The literature in the DSS field reported the effectiveness of various methodologies and tools (Vanhee & Lenstra 1994; Sprague 1980; Liberatore & Stylianou 1993; Hochman, Pearson & Litchfield 1994; Barr & Sharda 1997). DSS implementation literature draws from areas such as computer science for technical design; management information systems (MIS) for system development processes; organisational behaviour and change theory for managing the introduction of new system, human factors and ergonomics for man-machine interface design; applied psychology for prescriptive system support; management science and other disciplines for development of models to be included in the DSS (Nazareth 1993). Based on DDM paradigm, DSS is designed to include three major components, a dialog subsystem, a database subsystem and a model base subsystem; or an interface subsystem, a knowledge subsystem and a problem processing subsystem. The three-component architecture is capable of managing data, fitting data into models, and providing methods to reach decisions (Angehrn & Jelassi 1994).

There has been a recent trend towards using graphical user interface (GUI) in DSS. The development of GUI for a DSS has been a necessary requirement for the dialog component (Rathnam & Mannino 1995). Recently available system design tools were also used to explore the re-engineering of past management models and the computer system frameworks used to deliver these models in order to improve model expansion and to apply them to new problems. These models and frameworks were integrated to software tools to form DSS for operational use of managers in the field. The object-oriented (OO) approach is still a main way which is applied for the design or redesign of the structure of a DSS. A series of OO development frameworks and methods have been widely proposed for DSS development (Muhanna 1993). Some OO development software or tools can support the whole process of analysis, design and implementation of DSS (Power & Saarenmaa 1995). Successful implementation of DSS has also been studied by researchers such as Turban (1996) through studying an implementation framework and an implementation process.

2.3.6 Challenges of DSS

Keen (1987) identified three particular areas of future research for the next decade in the DSS field:

- problem identification;
- use of the multi-criteria methodology;
- shift of attention from individual decision to the collective one.

Sprague and Hugh (1995) indicated that the following challenges have to be met if this vision of DSS is to be realized:

Integrated Architecture

A common dialog interface, GUI, will allow access to all the information resources and previously separate systems, databases and model bases can be called and run from this common interface, where the application and data must be compatible enough to be accessible from this window.

Connectivity

Connectivity means the actual ability to connect workstations through local or wide area networks. It also requires a band width or data transfer rate to accommodate the interchange of data.

Document Data

The well-structured data in database has long been valuable for decision support. Even more important may be the concepts and ideas contained in less well-structured documents. Technologies are emerging that will allow access to, and management of, documents in addition to data records. This vastly increased set of information resources will have a major impact on the strength and effectiveness of DSS.

More Intelligence

Modelling and AI is the core of advanced DSS (Radermacher 1994). As expert systems began to develop and be used, some thought that they would replace DSS. DSS builders realized, however, that expert systems arts enhanced the forms of models, but if used only on a stand-alone basis, they suffer from the same limitations as stand-alone management science models. Thus, they were added to the model base and the DSS was used as a delivery system for expert systems.

2.4 Multiple Objectives Decision Support Systems (MODSS)

2.4.1 Integration of MCDM and DSS

DSS and MCDM models are generally aimed at supporting ill-structured decision making and both DSS and MCDM seek to support all phases of decision making, although the two disciplines are different in offering in relative support roles and have differences in support mechanisms (Nazareth 1993). The similarity of decision making problems addressed by the fields of DSS and MODM would suggest that they could borrow and build from each other. A marriage between DSS and MCDM promises to be practical and intellectually fruitful (Keen 1987).

Most DSS provide explicit support for the intelligence and design phases, and most DSS allow for active exploration of the problem space, with some evaluative capability of the solutions of interest. However, the choice phase of decision making is generally given little attention and tangible support. Likewise, the majority of MCDM models seek to prescribe a solution strategy during the choice phase primarily, and provide explicit support for evaluation and choice activities, with less emphasis on the generation of alternatives during intelligence and design phases of decision making. The selective use of their relative strengths would permit a more comprehensive mechanism for supporting the decision making process (Nazareth 1993). Therefore, an integration of MCDM and DSS ----MCDM based DSS (MCDMDSS) or in short multiple criteria decision support system (MCDSS) was proposed and considered by Jelassi et al. (1985) as a 'specific' type of system within the broad family of DSS.

The emergence of MCDSS was predicted in the early 1980s (Zeleny 1982). MCDSS was researched as a branch of DSS since the mid-1980s, such as by Jelassi (1985), Bui (1984), among others. Integration of MCDM into DSS had long been advocated by the researchers in both areas, but there were few examples of truly integrated working systems that possess all the functionality of DSS, yet provided the rich support for decision making offered by MCDM before early 1990s (Nazareth 1993). Nazareth (1993) examined the reasons for the lack of ready adoption of ideas, concepts, and proven strategies of each discipline in the complementary discipline. Since early 1990s, MCDSS started to gain more widespread attention (Korhonen and Wallenius 1990; Eom 1989; Eom and Min 1992; Nazareth 1993). And a set of tools and real systems of MCDSS which possess more functionality of DSS and MCDM has been developed (see Quaddus and Klass 1998; Klimberg 1992; Colorni et al. 1999; Korpela and Tuominen 1996; Tabucanon et al. 1994; Yau and Davis 1994; Poh et al. 1995; Iz 1991; Antunes et al. 1994). It has been recognised that interactive solution methods should provide not only a solution to a MCDM problem, but also the opportunity for decision makers to learn about their own preferences (Buchanan 1994). A series of studies in the 90's, such as Iz and Krajewski (1992), Eom (1996), Eom and Min (1992), Iz (1991), Limayem and Chelbi (1997), Angehrn and jelassi (1990), Antunes et al. (1994) reached the compelling conclusion that the MCDSS have positioned themselves at the core of DSS.

Even though MCDSS include much the same components as 'traditional' DSS, MCDSS have special characteristics that distinguish them from other DSS (Korhonen and Wallenius 1990; Korhonen et al. 1991). That is, MCDSS have a number of unique features that distinguish them from the features of classical DSS. Such characteristics include the fact that:

- they allow analysis of multiple criteria (objectives or attributes);
- they use a variety of multiple criteria decision methods (models) to compute efficient solutions;
- they incorporate users' input (interaction) in various phases of modelling and solving a problem.

The most important feature of MCDSS is the third one. The decision makers can make interaction in various stages of model management, model development and problem solving. MCDSS intend to provide the necessary computerized assistance to decision makers to solve multiple criteria decision problems. The decision makers are encouraged to explore the support tools available in an interactive fashion with the aim of further defining the nature of the problem. The ultimate success of DSS lies in its ability to help decision makers solve ill-structured problems through direct interaction with analytical models (Eom 1998). Such an ability can be enhanced by combining the various features of MCDM with DSS (Jelassi, Jarke & Stohr 1985; Jarke, Jelassi & Shakun 1987).

However, there is a long-term procedure to identify the concept of MCDSS. At first, when viewed from the DSS perspective, DSS researchers tend to treat MCDM models as any other model, and include them within the model management component of the DSS. On the other hand, MCDM researchers view an MCDM model as the DSS itself (Bui 1984, Zeleny 1982), in this case termed an MCDMDSS. Recently, there has been a trend towards constructing these MCDSS from a DSS perspective wherein explicit attention is paid to database management (Jelassi, Jarke & Stohr 1985), model management (Eom et al. 1987), and the user interface (Bui 1987; Korhonen and Wallenius 1990a; Klimberg 1992). In MCDSS, MCDM complements DSS and vice versa due to the differences in underlying philosophies, objectives, support mechanisms, and relative support roles. MCDSS, as the integration of DSS and MCDM, is construed to be the application of ideas, concepts, and strategies initially developed in one area, to problems better addressed in the other domain (Nazareth 1993). Researchers of both areas have accepted this standpoint.

As the separate areas, MADM and MODM tend to draw from different sources for their solution procedures. MCDSS can thus be broadly categorized into a generalized data-oriented MCDSS, which is based on MADM models, called multiple attribute decision support system (MADSS), and a model oriented MCDSS, which is based on MODM models, called multiple objectives decision support systems (MODSS). The two broad categories of MCDSS have different requirements in order to develop the respective systems, have different requirements of data and model management for effective decision support, and have different elements of the multiple criteria methodology matched with practical experience about the use of multiple criteria methods in MCDSS (Minch and Sanders 1986; Balestra and Tsoukias 1989).

2.4.2 Multiple Attribute Decision Support Systems (MADSS)

MADM has been presented earlier by equation (2.1) in Section 2.2.3. MADSS architecture has been proposed in the literature for example see Jelassi et al. (1985); Goicoechea (1989); Te'eni and Ginzberg (1991); Nazareth (1993) and among others. Various MADSS, that employ one or more MADM methods or multiple attribute analysis technique for determining the alternatives, have been developed, such as HIVIEW (Barclay 1987), EQUITY (Barclay 1988), PROMETHEE (Bois et al. 1989), TSDSS (Yau and Davis 1994), ALLOCATE (Quaddus and Klass 1998) and IMADS (Poh 1998); among many others.

The model management activities and data management functions are two of the most important characteristics of an MADSS (and MODSS). The model management function of the MADSS concentrates more on the 'problem structuring' and 'model structuring' functions. Watson and Buede (1987) defined the problem structuring as "the identification of the decision makers, the determination of decision making boundaries, the determination of the principal objective, the willingness and ability of other decision makers to provide inputs to the analysis". The problem structuring function can be integrated with the MADSS for better problem structuring support. According to Watson and Buede (1987), the model structuring involves "the creation of an evaluation tool for comparing the alternatives. Here the analyst must take the decision maker's objective and structure a value system".

A variety of MADSS can be found in the literature. Some earlier developed systems have a weak integrated function of DSS, a simple model management ability and data management ability. Some recently developed systems achieved improvements in the implementation of model management, data management, intelligent support, GUI and multiple decision makers environment, and have become a true MADSS. Table 2-1 surveys 15 various typical MADSS tools (generator), prototype and applications. Based on the principle of DSS, Table 2-1 reviews the characteristics related to:

- Does the MADSS have a methodology base?
- Does the MADSS have a database or similar data management ability?
- Does the MADSS have a knowledge base or other forms of an intelligent guide?
- Can the MADSS work in a group environment?
- Does the MADSS use GUI as the user interface?
- Is the MADSS generic (common) tool or a special application system (prototype)?

Brief description of the systems care provided below:

Table 2-1 List of MADSS

No	Name	Type	Method base	Data base	Intelli gent	Group	GUI	Reference
1	PREFCALC	C	N	N	N	N	N	(Lagreze & Shakun 1984)
2	HIVIEW	C	N	N	N	N	N	(Barclay 1987)
3	EQUITY	S	N	N	N	N	N	(Barclay 1988)
4	VISA	C	N	N	N	Y	Y	(Belton & Vickers 1989)
5	CRITERIUM	C	N	N	N	Y	N	(Sygenex 1989; Bois et al. 1989)
6	PROMETHEE	C	N	N	N	N	Y	(Bois et al. 1989)
7	MCDM advisor	C	Y	N	Y	N	Y	(Korhonen, Lewandowki & Wallenius 1991)
8	EXPERT CHOICE	C	N	N	N	Y	N	(Expert-Choice-Inc 1992)
9	GRADS	S	N	N	N	N	Y	(Klimberg 1992)
10	MCDA-DSS	C	Y	Y	N	N	Y	(Antunes et al., 1994)
11	TSDSS	S	N	N	N	N	Y	(Yau & Davis 1994)
12	MCView	C	N	Y	N	N	Y	(Vetschera & Rudolf 1994)
13	ICDSS	S	N	Y	N	N	Y	(Agrell 1995)
14	InterQuad	C	N	N	N	N	N	(Sun & Steuer 1996)
15	IMADS	C	Y	Y	Y	N	N	(Poh 1998)
16	ALLOCATE	S	N	N	N	N	Y	(Quaddus & Klass 1998)

Y---Yes
N---No
C---Common decision purpose tool
S---Specific task is addressed

PREFCALC (Lagreze & Shakun 1984) is based on multiple attribute additive value function and initial preference judgments on some reference alternatives. The value function of individual attribute here could be non-linear and various value functions are suggested which can be adjusted by the users to match their choice behaviour. The software is not highly user friendly and it is not very flexible either, but methodologically it is more elegant than other approaches.

HIVIEW (Barclay 1987) was one of the most widely used MADSS. HIVIEW was developed at the London School of Economics. It is based on a linear additive value model and has excellent graphic features. HIVIEW is excellent in model structuring and highly flexible in its use. One of the important features of HIVIEW is its ability to do various sensitivity analyses that will definitely help the decision makers to investigate various "what-if" type queries. HIVIEW also provides an inside view of different pairs of alternatives which becomes a learning process for the decision makers.

EQUITY is another MADSS which also has been developed at the London School of Economics (Barclay 1988). It is also based on a linear additive value model and quite simple to use. EQUITY is a dedicated software for equitable allocation of resources in a number of areas based on multiple attributes. The package provides a graphical display of efficient allocations and, by interactions with the decision makers, finds the most satisfactory allocation patterns.

Visual Interactive Sensitivity Analysis (VISA) (Belton & Vickers 1989) is also an MADSS which is based on a linear additive value model. It is useful for either a single user or a group of users. As the name implies, the most important feature of the software is its graphic based sensitivity analysis (both static and dynamic) which helps to clarify various fuzzy and uncertain issues of the problem.

Another MADSS based on a analytic hierarchy process (AHP) is called CRITERIUM (Sygenex 1989). It is also excellent for problem structuring. It has excellent facilities to report and document the decisions. It can also be used for either single or group situations.

PROMETHEE (Bois et al. 1989) is an MADSS based on an outranking method. However it uses a kind of preference function to find the equivalent of a concordance index and discordance index. The software provides a complete ranking of the alternatives and displays it graphically.

The MCDM advisor is an expert system based system and has a methodology base of 8 MADM methods. The expert system helps the decision makers to find the appropriate method to be used (Korhonen, Lewandowki & Wallenius 1991).

One of the most widely used MADSS is called EXPERT CHOICE (Expert-Choice-Inc 1992). It is based on the theory of AHP. The software is excellent in problem structuring in a hierarchical fashion. It has various options in judgment elicitation e.g. verbal, numeric and graphic. One of the main features of the software is its various modes of sensitivity analysis e.g. gradient, dynamic and performance. The package has been widely used to analyse thousands of planning problems both in individual and group situations.

GRADS (Klimberg 1992) is an MADSS based on visual displays. GRADS is not intended for 'optimal' decision making. It merely supports the human choice behaviour by graphical displays. It is highly useful for the decision makers who find it difficult to aggregate the multiple criteria in any form.

Antunes et al. (1994) developed an MADSS which consists of a problem editor, a problem compiler, a set of multiple criteria decision aid (MCDA) methods, a dialogue base and a data management module (it is called MCDA-DSS in Table 2-1). The MADSS intends to make the most of a user-friendly computer interface in order to minimize the cognitive effort required from decision makers. The set of methods include some presentative discrete alternative MCDA methods, underlying different philosophies and strategies for selecting, clustering or ranking the alternatives. This MADSS follows an experimentation framework where the user can try different methodologies and freely change his/her preferences in order to evaluate different scenarios, in an interactive manner. Due to its modular structure this MADSS can easily create new methods and various graphical interfaces.

Yau and Davis (1994) developed a tenant selection decision support system (TSDSS) which employed a multiple attribute analysis technique as its evaluation tool. TSDSS also introduced a new method for determining the alternatives to be considered in the selection process, so that the values of some criteria, which represent the characteristics of the combinations of the chosen tenants, can be obtained.

MCView (Vetschera & Rudolf 1994) is a DSS for multiple attribute decision problems. The system combines a process-oriented view of multiple attribute decision making with a comprehensive GUI. The decision process supported by MCView is based on a two-level representation of evolving preferences, which allows the user to interactively introduce preference information both in the form of holistic choices between alternatives and by directly changing the parameters of a cardinal evaluation method. The GUI of MCView provides comprehensive information about and direct manipulation of all problem components and thus directly supports this learning process.

Agrell (1995) presented an interactive multiple criteria framework for an inventory control decision support system (called ICDSS in Table 2-1). The decision support procedure is constructive in the sense that the preference structure of the decision maker is assessed progressively under the exploration of the solution space. The framework has been intended for inclusion in a DSS for production and operations management or to be used as a separate module for strategic inventory control. Implementations as FORTRAN modules and spreadsheet macros are available.

InterQuad (Sun & Steuer 1996) was designed with large discrete alternative problems based on a data structure called a quad tree. InterQuad takes advantage of the ability of a quad tree to identify, store, and retrieve non-dominated criterion vectors. The user interacts with the non-dominated criterion vectors stored in the quad tree.

IMADM was developed by Poh (1998). This is an intelligent multiple attributes decision making system prototype. In this prototype, 11 well-established methods, including ELECTRE, TOPSIS, MAXIMIN, SAW, AHP, have been implemented. A knowledge-based system (KBS) has been utilised to provide guidance on the selection of suitable methods for MADM problems. The knowledge–base guidance system works by depending only on inputs from the users and utilizes a knowledge base to determine the best MADM method that matches the needs of the specific problem. Inputs elicited from the users to the guidance system include the problem nature, data or information availability, and the desired nature of the solution. To provide appropriate guidance for users possessing different levels of expertise, three modes of guidance had been incorporated. IMADM was implemented in C++ and used an expert shell CLIPS.

ALLOCATE (Williams, Klass & Morien 1997, Quaddus & Klass 1998), is another MADSS which also deals with allocation of resources among competing projects based on a linear value model. In principle, ALLOCATE is very similar to EQUITY. However ALLOCATE is more flexible and deals with both cumulative and independent sets of projects. It also has some innovative graphic features that help the decision makers in arriving at a compromise solution. It was implemented in DELPHI.

It must be mentioned that this section only provides the brief overview highlighting the salient features. The details are available from the cited references. It is also noted that this section doesn't evaluate the advantages and disadvantages of the MADSSs in Table 2-1. For successful applications, a number of factors, including the problem situation, availability of data, target groups, the decision makers and various other bureaucratic and social-political situations have to be considered.

2.4.3 Multiple Objectives Decision Support Systems (MODSS)

MODSS is applied to support the decision making problems which involve multiple objectives, and there exist conflicts between these objectives. Mathematical representation of MODM is provided by equation (2.2) of Section 2.2.4. MODSS has gained widespread attention in the algorithms (Hwang and Masud 1979; Steuer 1977; Hwang and Yoon, 1981) and methodologies implementation (Korhonen and Wallenius 1990b; Shin and Ravindran 1991; Poh et al. 1995) as well as in their applications (Lawrence et al. 1997; Shahsavar et al. 1995; Stewart 1991; Yakowitz and Hipel 1997; Yakowitz et al. 1993; Werners 1987; Ogryczak et al. 1989).

Compared to MADSS, MODSS requires more model management functions than data management functions, although its model structuring is difficult. The problem structuring function of the MODSS should be very similar to the MADSS. Problem structuring in MODSS mainly generate the 'objectives', 'constraints' and 'decision variables', three of the most important components of the MODM model. The model structuring functions of the MODSS has to be performed initially by the decision maker. The model structuring, therefore, should be done manually first and then must be evolved gradually over a number of iterations (Korhonen, Moskowitz & Wallenius 1992). The model can be structured in an interactive fashion using GUI environment. Two matrices, called 'objective matrix' and 'constraint matrix', must be generated first to construct the models of MODM (Quaddus & Siddique 1996a; Quaddus 1997).

Unlike MADSS, there are not many MODSS reported in the literature. Like MADSS, a review of these MODSS reveals that many of them are only computer applications of MODM, very few of them would satisfy the requirements of a DSS.

Table 2-2 surveys 12 MODSS tools (generators), prototypes and applications from the literature and reviews the characteristics related to the same issues of MADSS (Table 2-1):

- Does the MODSS have a methodology base?
- Does the MODSS have a database or similar data management ability?
- Does the MODSS have a knowledge base or other forms of intelligent guide?
- Can the MODSS work in a group environment?
- Does the MODSS use a GUI as user interface?
- Is the MODSS a generic support tool or a specific application system (prototype)?

Table 2-2 List of MODSS

No	Name	Type	Method Base	Data base	Intelligent	Group	GUI	Reference
1	IMOP	C	Y	N	N	N	N	(Werners 1987)
2	DINAS	S	N	N	N	N	N	(Ogryczak, Studzinski & Zorychta 1989)
3	VIG	C	N	N	N	N	Y	(Korhonen & Wallenius 1990a)
4	HYBRID	C	N	N	N	N	N	(Poh & Quaddus 1990)
5	Interactiv e MOLP	C	Y	Y	N	N	Y	(Korhonen, Lewandowki & Wallenius 1991)
6	R&DPS MODSS	S	N	Y	N	N	Y	(Stewart 1991)
7	WQ-PMODSS	S	N	Y	N	N	Y	(Yakowitz et al. 1993)
8	ISGPII	C	N	N	N	N	Y	(Hwang et al., 1993)
9	MOLP-PC	C	Y	N	N	N	N	(Poh, Quaddus & Chin 1995)
10	IMOST	C	Y	Y	Y	N	Y	(Lai 1995)
11	PDSS	C	N	Y	N	N	Y	(Paige et al. 1996a)
12	FORMDS S	S	N	Y	N	Y	Y	(Tecle, Shrestha & Duckstein 1998)
13	GMCRII	C	N	N	N	Y	Y	(Hipel et al. 1998)

Y---Yes
N---No
C---Common type of decision purpose tool
S---Specific task is addressed

Brief descriptions of the systems are provided below.

Interactive Multiple Objective Programming (called IMOP in Table 2-2) was developed in Werners (1987). The interactive DSS used the aids to solve multiple objective programming problems subject to both strict and flexible constraints. An integral part of the system is an extension of a well-known fuzzy sets approach assessing possible solutions by their degrees of membership to objectives and constraints. This procedure is linked to classical multiple objective programming models. If the decision maker cannot determine membership functions a priori, the system suggests functions dependent on the given information and permits interactive modifications. In addition to the graphics, the MODSS contains considerations about empty or unconstrained solution spaces and backtracking opportunities to former solutions.

The Dynamic Interactive Network Analysis System (DINAS) was researched, developed and implemented by Ogryczak et al. (1989). DINAS enabled the solution of various multiple objective transhipment problems with facility location. DINAS used an extension of the classical reference-point approach to handle multiple objectives. In this approach, the decision maker forms the requirements in terms of aspiration and reservation levels, that is, the decision maker specifies acceptable and required values for given objectives. A pilot version of DINAS was implemented on an IBM PC/XT as a menu-driven and easy-to-use system armed with a special network editor.

Korhonen & Wallenius (1990b, 1990a, 1991) developed an MODSS called Visual Interactive Goal Programming (VIG). The system is user-friendly and is designed to support both the modelling and solving of a multiple objective linear programming problem. The VIG computer program was written in TURBO PASCAL. An essential component of VIG is PARETO RACE, a visual, dynamic search procedure for exploring the efficient frontier of a multiple objective linear programming problem. VIG was based on goal programming and had interactive graphic features which helped the decision makers to investigate various solutions of multiple objective linear programming problems.

HYBRID was developed by Poh and Quaddus (1990). HYBRID uses the solution of a two-person zero-sum game with mixed strategies to generate efficient solutions, and then proceeds to modify the feasible region using responses from the decision maker. The cycle is repeated until a satisfactory solution is obtained. This method was implemented using TURBO PASCAL.

The Interactive MOLP of Climaco and Antunes (Korhonen, Lewandowki & Wallenius 1991) is an interactive-based MODSS. It has a MODM methodology base of five popular multiple objective linear programs (MOLP) methods. These methods can be used in stand alone mode or in any sequence the user wishes.

Stewart (1991) described the development of an MODSS for the selection of a portfolio of R&D projects, which was carried out for a large electricity utility corporation. The MODSS (called R&DPS MODSS in Table 2-2) was constructed around a reference point approach for the underlying multiple objective decision problems.

A prototype multiple objective decision support system for water quality (`called WQ-PMODSS in Table 2-2) has been developed by Yakowitz et al. (1993). The MODSS used a simulation model to predict the impact of alternative management systems on surface and groundwater quality as well as farm income. The MODM component of the MODSS made use of dimensionless utility functions and the concept of importance order dominance. The MODSS allowed the user to set up alternatives, choose the decision criteria, and accept or customize the decision mode settings. The results were presented graphically as well as in tabular form.

ISGP-II is an interactive DSS to solve a multiple objective decision making problem by Hwang, Lai & Ko (1993). ISGP-II provided a process of psychological convergence for the decision maker, whereby it learns to recognize good solutions and their importance in the system, and to design an optimal system, instead of optimising a given system.

MOLP-PC is an integrated MODSS which has been applied to a number of planning problems (Poh, Quaddus & Chin 1995). The best feature of MOLP-PC is that it has a methodology base of fourteen popular MODM methodologies, including IMOLP, ESGP, LGP, Step Method, STEUER and ISGP, for solving multiple objective linear programs (MOLP). MOLP-PC was written in TURBO PASCAL and was implemented as a DOS application. It was centred on a customized 'spreadsheet' on which MOLP models were represented and manipulated. MOLP-PC is highly user friendly but it does not yet have graphic facilities for better interaction with the decision makers.

An interactive multiple objective system technique (IMOST) was investigated by Lai (1995) to improve the flexibility and robustness of MODM methodologies. The interactive concept provided a learning process about the system, whereby the decision maker can learn to recognize good solutions, the relative importance of factors in the system, and then design a high-productivity and zero-buffer system instead of optimising a given system. This system was implemented with the help of the computer package LINDO, which can be used to solve an integer linear programming problem.

A prototype decision support system (PDSS), using multiple objective decision theory and embedded simulation models, was developed to evaluate landfill cover designs for low level radioactive waste disposal sites by Paige et al. (1996a). PDSS applied the decision model based on multiple objective decision theory and used a unique approach to order the decision variables and rank the design alternatives. The PDSS has been evaluated using the Hill Air Force Base landfill cover demonstration project (Paige et al. 1996b).

Tecle et al. (1998) developed a multiple objective and/or multiple person DSS for analysing multiple resource forest management problems, called FORMDSS. The procedure includes formulating the problem in a multiple objective and group decision making framework, and solving it using two solution techniques which consist of a distance-based compromise programming and a cooperative game theoretic approach of the Nash equilibrium type. The software was designed to provide natural resources management decision makers with an interactive, flexible and easy to handle DSS.

GMCR II was developed by Hipel et al. (1998) as an MODSS tool for providing strategic advice in multiple participant-multiple objective decision making situations. The MODSS can be utilized by managers and consultants for better understanding a conflict situation.

It has been found that there are no (to the best of authors knowledge) MODSS which supports the complete model management function with a methodology base, data management function with a database and an intelligent guide with a knowledge base, which can be applicable for both single and group situation with GUI technology as described above. For example, MOLP-PC possessed the model structuring support with a methodology base, but not an intelligent guide for problem structuring and only can be used for a single decision maker. While FORMDSS can be applied for a multiple objective and/or multiple person decision problems, but only for analysing multiple resource forest management, it doesn't have a methodology base and an intelligent support.

It must be noted that many of MADSS and MODSS (all Table 2-1 and Table 2-2) are not 'truly' DSS. They really are only software implementations with some interactive features.

It must be also noted that some other MADSS and MODSS are available in theory and practice although they are not in Table 2-1 and Table 2-2. For example, Kim, Ahn & Choi (1997) examined the application of multiple objective linear goal programming in military budget planning, Hipel (1992) has applied multiple objective decision making methods in water resource management, Tzeng and Chen (1993), and Tzeng et al., (1994) have an MODM application for traffic assignment and an application to deal with energy supply mix problems, among many others.

2.5 Intelligent Multiple Objectives Decision Support Systems (IMODSS)

2.5.1 Integration of Artificial Intelligence (AI) and DSS

Artificial intelligence (AI), that is:

- knowledge-based/expert systems (KBS/ES),
- natural language analysis, and
- search, inference and planning procedures

has experienced significant progress in research and implementation (Pau 1986). As a powerful tool, AI allows a human-being to easily control and direct power sources in the accomplishment of a task by providing cognitive amplification or augmentation. Economists, banks, financial services, as well as many management departments are paying increasing attention to this surge of activity (Espinasse & Nabitz 1993; Pau 1986; Spangler 1991).

Research in AI generally falls into one of two major categories (Fordyce & Sullivan 1986):

- making machines more useful to humans; and
- understanding intelligence.

Since the focus of this thesis is about knowledge-based intelligent guidance for decision making, it deals with the implementations of AI which functions as tools in support of the decision maker, which belongs to the 'understanding intelligence' category. As this focus specifically deals with the combination of DSS and expert systems (ES) sub-area of AI, the literature in ES technology, integrating DSS with ES, and knowledge-based ES will be mainly reviewed here.

A knowledge-based expert system is defined to be a class of computer programs that uses knowledge and problem-solving techniques on a skill level comparable to those of human experts and intends to serve as consultants for decision making (Rauch 1984). These systems consist of a knowledge base, containing facts, rules, heuristics, and situation patterns, and an inference system that makes decisions within a domain. They differ substantially from conventional computer programs in that their goals may have no algorithmic solution, and they must make inferences based on incomplete or uncertain information. They are called expert systems because researchers have found that amassing a large amount of knowledge, rather than sophisticated reason techniques, is responsible for the success of the approach (Pau 1986).

Knowledge-based expert systems are one of the key tools in moving DSS into the next step in the evolution from passive data storage to highly active systems that participate in the decision making process (Rychener 1985). Knowledge-based expert systems enable information system builders to move problem domain knowledge from the human to the computer so as to support problem recognition, problem structure and problem solving (Klein & Methlie 1995). Knowledge-based expert systems play two different roles, supporting and replacing a decision maker, at different organisational levels (Edwards, 2000). The growing interest in knowledge-based expert systems is attributable to the many potential benefits they can provide, including (Pau 1986):

- the ability to codify human expertise permanently;
- wider accessibility to that precious expertise;
- a second opinion to that of a practicing expert;
- the opportunity to explore 'what if...' situations with different types of inputs;
- the ability to handle uncertainty when data are incomplete; and
- the ability to solve problems that have extremely large sets of possible solutions to explore.

The complexity of decision making is increasing. The active involvement of the user and the computer in an intelligent way is necessary in decision process. Knowledge-based expert systems provides expertise when human expertise is not available, provide expertise more uniformly, and sometimes faster and assist experts in making decisions in complex situations. It has become a trend that DSS products incorporate, and will eventually encompass, tools and techniques from AI, particularly from ES. The tools of ES are an emerging technology that DSS developers recognize as both a key challenge and a key puzzle (Keen 1987). The knowledge base has become a form of combined data/model base, the inference engine can be viewed as a knowledge base management system (similar to the data base management system and dialog management system), and the language system is a part of the dialog.

Zanneto, as far back as 1968, had already foreseen some of the properties of future MIS using AI technology (Zannetos 1968). However, his paper deals mainly with MIS and not DSS. In the middle of 1980's, a conference that was held in 1984 on expert database system (EDBS), primarily involving the integration of database and ES. Ford (1985), Sen and Biswas (1985), Turban and Watkins (1986), Turban and Sepehri (1986), Garsombke and Parker (1987), Holsapple and Whinston (1987), Henderson (1987), Van and Lapinski (1988), Liebowitz (1988) as well as other researchers studied the integration of ES technology within the DSS framework. For example, Ford (1985) compared DSS and ES, and found that both commonly use an iterative design or prototyping development approach. Some ES concepts may be transferable to DSS development. ES may have many business applications, and the application of both systems is likely to increase, as will their impact on decision analysis and decision making. Turban and Watkins (1986) proposed two kinds of possible connections between the DSS and ES. These are:

- ES integration into the conventional DSS components; and
- ES as an additional component of DSS.

A variety of technical, behavioural, and design issues were discussed that must be addressed in the DSS-ES integration. ES can make DSS a more active and potentially more valuable part of the decision process. The DSS-ES integration not only answers what if questions, it also will be able to answer why. Rao and Lingaraj (1988) discussed ES in production and operations management and provided their classification. In the early 1990s, the studies about these combinations have received great attention e.g. Finlay (1990) published the paper titled 'Decision support systems and expert systems: a comparison of their components and design methodologies'. This research contrasted and compared DSS and ES, highlighted the large areas of similarity between DSS and ES, and indicated to operations research (OR) practitioners that the training and experience of OR specialists are such that they are capable of playing a significant role in the development of ES. Edwards (1992) again indicated that although the fields of DSS and ES grew up in different backgrounds - MIS and AI, respectively, operational researchers are working in both of them. Other researches can be found from the literature, such as Connell and Powell (1990), Higby and Farah (1991), Despres and Rosenthal (1992), Pomerol (1993), Gan and Yang (1994), Rao et al. (1994), Liberatore and Stylianou (1993). These researchers have focused on tandem architectures that synthesize ES and DSS.

An ES is a decision-making/problem-solving integrated package that contains a knowledge base for a particular problem domain, and a reasoning mechanism for inference (Rao & Lingaraj 1988; Rao, Sridhar & Narain 1994). A DSS on the other hand helps decision makers utilize data and models from its database and model-base to solve unstructured or semi-structured problems by allowing the user to approach a problem analysis in a flexible, personal way (Ford 1985). While a typical DSS supports quantitative, mathematical, and analytical reasoning, a typical expert system can support qualitative analysis based on methodologies such as symbolic reasoning and pattern recognition. Integrating the two into an intelligent decision support system (Ford 1985; Remus & Kottemann 1986) can thus support various types of problem solving processes and provide active modes of support to the decision maker. In the middle of 1990s, more research has been done. For example, Yang (1995), Degoulet (1995) discussed DSS from the standpoint of knowledge representation. The Relationships between DSS and knowledge representation are examined from three different points of views. Owens and Philippakis (1995) provided inductive consistency in knowledge-based DSS (KBDSS). Efforts to integrate knowledge into DSS have lead to approach that incorporates induction techniques to derive rules from data. Such integration efforts create a new integrity control challenge for KBDSS. More recently, a literature search was done covering various journal and conference publications for the period 1985-1995 for materials on tools for the validation and verification of knowledge-based systems (Murrell & Plant 1997). A variety of researches have indicated the components of knowledge-based expert systems that can be successfully integrated into DSS and thus growing role of this integrated KBDSS or intelligent DSS (IDSS) has occurred. The future will bring much more integration and extension of the intelligence capability in DSS (Kasper 1996).

2.5.2 KBDSS and IDSS

The role of the KBDSS or IDSS is not to replace the human decision makers, but to function as a tool for decision making by complementing the decision makers abilities of problem solving in the application domain. In the process of problem solving, at various intermediate stages, the IDSS presents a decision maker with different alternatives to choose from, thus determining the problem solving path. By providing a user with the ability to determine the problem solving path, the IDSS has the flexibility of adapting to a user's style of problem solving, an important and essential aspect in the design of DSS (Rao, Sridhar & Narain 1994). IDSS provides the mechanism for assimilation of ES, knowledge representation, natural language query, pattern recognition and so on. The IDSS can 'suggest', 'learn' and 'understand' in dealing with managerial tasks and problems.

The advent of improved techniques in computer system development and AI makes it feasible to implement active IDSS (Rao, Sridhar & Narain 1994). Development and implementation of IDSS to support intelligent decision making is an area of research that has gained in importance in recent years. Various IDSS applications have also been developed. For example, Budgen and Marashi (1988) designed MDSE advisor where knowledge-based techniques applied to software design assessment. Reitman (1990) developed two generic expert systems for management applications: the operations advisor (OA) and the management advisor (MA). Venkatachalam (1999) developed an intelligent model selection and forecasting system which is based on neural network technology for model selection and forecasting. Shan and Xu (1996) provided a KBDSS called KB-CEDSS that is integrated with a multi-layer artificial neural network (ANN) for evaluating urban development. Borenstein (1998) presents IDSSFLEX, an intelligent decision support system, which selects from among several configurations and control strategy alternatives of design, the most appropriate one for a specific case in an individual environment. Seng (1996) explored a new approach to build interactive IDSS. The integrated system combined the strength of rule-based semantic structure and the learning capability of connectionist architecture. In addition, the system allows users to define logical operators that behave similarly to that of the human expert decision making process. For other published research on IDSS, see Morgan (1996), Poh (1998), Chen (1998), Degoulet et al. (1995), Edwards et al. (1995), Gottinger and Weimann (1995), Kahn (1993), Pau (1986), Shahsavar et al. (1995), Spangler (1991) and Taha et al. (1995), among manes others.

2.5.3 Intelligent Multiple Criteria Decision Support Systems (IMCDSS) and Intelligent Multiple Objectives Decision Support Systems (IMODSS)

As discussed above, DSS has provided convenient means and tools for decision makers to apply MCDM methodologies to their fullest potential. Most of the MCDM methodologies that have been developed could be implemented as computer software or embedded in a DSS. Unfortunately, the complex mathematics involved in many of these MCDM methodologies deters many corporate decision makers, who are not formally trained in the field of MCDM, from utilizing them. In addition, the large number of methods available posed an additional level of complexity to unfamiliar users who were unable to determine the most effective method or methods for any given specific problem (Poh 1998). This is due to their

- lack of expertise and experience to understand specific features of an available MCDM;

- lack of experience to systematically analyse and decompose complex problems into subtasks; and
- lack of ability to match the most appropriate MCDM techniques with current problem subtasks.

Oftentimes, the assessment of the selection factors relies much on the knowledge of one or many human MCDM experts or consultants. This is particularly true when the alternatives and /or their evaluation criteria have to be modified to reach a convergent and stable outcome. It would thus be useful to store this expertise in a set of knowledge bases to assist the novice to casual MCDM users (Bui & Sivasankaran 1988).

Many papers have presented how MCDM and ES for decision support had been combined in the past and can be integrated in the future to produce problem solving systems that are sensitive to the key role played by user preference in the problem-solving process. A survey by White (1990) has provided insights as to how system capability and acceptability can be enhanced by this integration.

As mentioned above, MCDM is concerned with applying multiple criteria decision theory and models to real-world problems, and as such is concerned not only with satisfying the axioms of decision theory but also with alternative, attribute, objective and consequence generation, probability and utility assessment, and search for an alternative or objective that maximizes expected utility (White 1990). An ES can be designed to emulate the problem-solving behaviour of an expert in the context of a complex, but usually well-constrained, problem. Combination of ES and MCDM will considerably increase the expertise embedded in solving MCDM problem and will improve the capacity for the decision makers to enhance this expertise (Klein & Methlie 1995).

A set of knowledge-based researches for MCDM has been reported in the literature. These researches focus various aspects of integrating AI (or ES) with MODM, including providing a theoretical basis for intelligent based MCDM, examining the relationship between AI, computer science, economics, psychology, operations research, and management science, outlining an approach to decision aid development that uses both the problem structuring approaches of MCDM and the incrementally modifiable software architectures found in AI, exploring how AI based decision aids treat preferences that are the driving force for making decisions. Others (Keeney 1988, Davies, 1994, Pereira and Duckstein, 1993). Particularly, Pomerol (1993) gave a more detail discussions about the combination of ES and MCDM. Their research evaluated the alternatives or constructed the utility functions. The study also imagined a structure in which each alternative is characterized by a fact base and each criterion by a rule base. The rule base is organised as a semantic tree of criteria. The evaluation of an alternative is made by applying the rules to the facts characterizing each alternative. The result of the system is a decision matrix.

An application framework was provided in Bui and Sivasankaran (1988). This framework described three kinds of applications of ES in the MCDM, particularly, MCDSS, that they are considered particularly fruitful.

First is the development of new computerized MCDM tools that combine existing MCDM techniques and knowledge-based methods. Such a system would provide an integrated and complete MCDM environment. However, its implementation requires substantial development effort.

Second is the area of advice-giving systems for managers inexperienced in using MCDM. Expert systems particularly lend themselves for user-training through their ability to perform rule traces of their inference process.

The third application advocated is that of an intelligent front -end for existing MCDM packages. This is an expert system that sits between a package and a user, generates the necessary instructions or code to use the package following a dialogue with the user, and interprets and explains results from the package.

With the architecture of applications, some integration and extension of the intelligence capability, with ES, as a guidance system, advice system or intelligent frond-end, have been brought into MCDSS. Some efforts have been made to integrate various ES and

AI techniques into the MCDSS to develop the knowledge-based or intelligent MCDSS (KBMCDSS or IMCDSS), further KBMADSS or IMADSS, and KBMODSS or IMODSS. The IMCDSS may guide and provide reasoning about the appropriateness of the MCDM model formulation (structuring a decision), exploration/construction of the alternative set, evaluation of the alternatives/criteria, construction of the utility functions, and interpretation of outputs under a DSS (Shahsavar et al. 1995;Eom 1998). McIvor et al. (1997) showed how knowledge-based systems technology can assist in the area of strategic purchasing. This knowledge-based system is specific on the issues involved in the application of multiple attribute analysis to the automation of the make or buy decision. An intelligent front-end for MCDM-based DSS was designed by Bui and Sivasankaran (1988). This rule-based front-end can activate independent MCDM programs that can be run conventionally. The authors showed how ES technology can be applied to increase effective use of MCDM. With the architecture of the intelligent front-end, it can achieve better integration of the different available models throughout the decision making processes. Furthermore, a suggestion was provided by Ko and Lin (1988) to utilize KBS to assist decision maker in formulating MCDM problems, to improve the situations, to provide a better user interface and to select the best MCDM methods in a given problem situation. Ko and Lin built a prototype expert advisory system for selecting MCDM model using the Personal Consultant Plus expert system shell. However, there was no MCDM methodology base integrated with KBS to provide users with a complete decision making system. A successful implementation of IMADSS was completed by Poh (1998). This is an intelligent multiple attributes decision making system prototype. In this prototype, a knowledge-based system is utilised to provide guidance on the selection of suitable methods for MADM problems. However, integration of the knowledge base techniques into an MODSS to support the selection of best suitable MODM method for a particular application has not been found in the literature review.

It is important to remark that ES's that support, and assist decision making are only concerned with one or a few aspect(s) of IDSS. Our emphasis in this Chapter is on ES's that support decision making for problems having a significant dependence on user preferences.

2.6 Multiple Objectives Group Decision Support Systems (MOGDSS)

2.6.1 Group Decision Making (GDM)

Maximization and satisfaction are two major frameworks for decision making. A combination of both may be used to describe GDM. In the satisfaction approach, decision makers formulate aspiration levels or demands that take the form of constraints. Choosing from among different decision alternatives, decision makers take into accounts their preferences or wants, which take the form of objectives or goals (Kersten 1985; Johnson 1991; Schwartz 1994).

Decision making groups are formed to exchange information and ideas, and to identify acceptable and desirable alternatives and solutions (Johnson 1991; Schwartz & Levin 1990;Anonymous 1988). The structured decision-making process is based on a method that includes putting ideas in writing, discussion of ideas, and ranking and discussion of priorities (Adizes & Turban 1985). However, many times decision groups reach a dead end due to the differences among individual interpretations of the 'best compromise' solution. Despite an initial agreement on a set of criteria, disagreements among decision makers may arise in determining a final solution to the multiple objectives decision problem. There is no rule for combining individual preferences into a group preference unless interpersonal comparison of utilities is allowed. Consequently, most utility group aggregation methods require explicit interpersonal comparisons of utility and follow a normative approach assuming that a group decision rule can be constructed by aggregating the utility functions of group members (Iz & Jelassi 1990).

Many decision making activities under a group may take one of actions below (Schwartz 1994):

- decision by lack of response,
- decision by authority rule,
- decision by minority rule, and

- decision by majority rule.

One of the best, but most time-consuming techniques, is decision by consensus, where communications are sufficiently open and the group climate has been sufficiently supportive to make everyone in the group feel that they have had their fair chance to influence the decision (Schwartz & Levin 1990).

Usually, the steps in a group decision are:

- identify the problem;
- clarify the problem;
- analyse the cause;
- solicit alternative solutions to the problem;
- select one or more alternatives for action;
- plan for implementation;
- clarify the contract;
- carry out the action plan;
- provide for evaluation and accountability.

Advances in computing and telecommunications technology are changing how people can meet and make group decisions. Technological changes help people cross physical, social, and psychological boundaries and have secondary effects on group behaviour and decision making (Valacich & Schwenk 1995). Groupware, which consists of computer software aimed at improving communication within teams of staff, is one of the categories of information systems being developed for use in a group. Commercially available systems offer business users a structure within which they can make decisions more quickly, with more input from a wide network of experts, and with vastly improved coordination.

Many firms involve employees in a GDM to facilitate change, as well as to ensure greater employee acceptance of new procedures and policies. Before implementing a GDM, the conditions necessary for its success should be considered (Anonymous 1988). Groups are more likely to succeed if certain conditions exist, such as when:

- a range of possible solutions is initially available for consideration,
- the need for personal expression is limited, and
- participants acknowledge the power of unified action.

Furthermore, to ensure success, management must fit the technique (interactive or non-interactive) to the problem at hand.

Researchers and practitioners have long been interested in the effects of GDM. Some widely used and studied techniques have been found to enhance decision making performance for group. GDM has been well researched in its theory such as Anonymous (1988), Weiss and Assous (1987), Driskell and Salas (1991), Sniezek (1992), Kiesler and Sproull (1992), Paese et al. (1993) and Hartman and Nelson (1996). Based on the theory research various of GDM approaches, such as Kerr (1992), Ward and Reingen (1990), Jacob and Pirkul (1992), Hiltz et al. (1989), Chung and Adams (1997), are exhibited in literature. Particularly, many types of group aggregation processes and group aggregation methods have been proposed in the literature. These include mathematical programming, AHP (analytic hierarchy process), weighting alternatives, performance weighting aggregation method and so on. Particularly, some GDM methods under multiple criteria have also been provided (Hwang & Lin 1987). Some were extended and analysed as computer-based aggregation method and have been implemented and embedded in DSS. How to use these approaches effectively in a group decision meeting and how to integrate these approaches into DSS so that people (particularly mid-level executives) do not require significant training? Some related experiments and application researches have been done such as Zigurs et al. (1988), Herriot et al. (1985), Corfman and Lehmann (1987), Dennis (1996) and Travica and Cronin (1995). Despite such theoretical development, experimental researches and design approaches in GDM and group preference aggregation, most of the real world applications in this area involve theoretically less rigorous but more practical aggregation procedures and need more easy-to-use tools (Iz & Jelassi 1990).

2.6.2 Group Support Systems (GSS)

Group activities may not always involve high-level decision making. In a more generic sense, GSS has become more commonly used to encompass all systems that support group work of any kind. GSS represents a new class of technologies and methodologies that have been developed to support group work and to improve the efficiency and effectiveness of group meetings. Numerous authors have described applications of GSS to group work in a variety of areas and many have proposed other applications which seem amenable to such support (Pendergast and Hayne 1999; Tung and Turban 1998; Wolfe 1994; Pervan 1996; Dennis 1996).

Though many types of GSS have been designed, two fundamentally different viewpoints have underpinned most of the approaches. One approach assumes that the task of a group is to exercise discretion, which implies that the support provided must allow the members of the group to consider uncertainty, form preferences, make tradeoffs and take decisions. This approach recognizes that most group decision making should rely on the application of modelling and decision theory, an understanding of group processes, and the use of information technology. This approach has been called Decision Conference (DC) (Phillips 1988) and one form of its practical implementation is exemplified in the use of multi-criteria modelling software such as Equity and Hiview (Pervan 1996). DC usually uses a single workstation operated by a skilled analyst or facilitator and the participants do not require hands-on access to the computer-based support. Other approaches that fit within this category include Strategic Options Development and Analysis (SODA) (Eden 1989), the Strategic Choice (Friend & Hickling 1987) and others. A second approach is driven by communication needs and utilises computer technology as a means of facilitating group communication. This approach assumes that interpersonal communication is the primary activity of group decision making and that the function of a GSS is to improve the group's communication (through the application of computer hardware and software). Systems supporting this approach are usually called Electronic Meeting Systems (EMS), they operate with each participant having a workstation, and examples include the Group Systems software from the University of Arizona (Dennis et al. 1988).

2.6.3 Group Decision Support Systems (GDSS)

DSS have been well researched and a variety of interactive solution methods of GDM have been derived. Systems that combine appropriate technologies, and methodologies of DSS and GDM show the potential to enhance the efficiency and effectiveness of group decision work. Such applications of information technology to support the decision work of groups have been referred to as GDSS (Gray 1987). A GDSS is characterized as an interactive computer-based information system that combines the capabilities of communication technologies, database technologies, computer technologies, and decision technologies to support the identification, analysis, formulation, evaluation and solution of semi-structured or unstructured problems by a group in a user-friendly computing environment (Huber 1984). GDSS typically offer a wide range of capabilities, including computerized support for interactive modelling, traditional group preference aggregation mechanisms, communication, idea generation, and public display of information (DeSanctis and Gallupe 1987; Er and Ng 1995). An important point to note is that the GDSS is used in decision group (not general group meeting) and to support decision making (not only creating alternatives). This is the basic difference between GDSS and GSS mentioned in Section 2.6.2. In some group meetings to creating and/or reviewing alternatives to be submitted as a short list to the next level in the organisational hierarchy, GSS is often used rather than GDSS (Sprague 1995).

Interest in the development of computerized GDSS emerged in the early 1980s (Gray 1987). For interdependent decision support, GDSS has become much more prevalent in the past few years. The growing availability of local area networks and group communication services, such as electronic mail, is making this type of DSS increasingly available. A variety of academic articles on GDSS have appeared in the literature (Bui, Jelassi & Shakun 1990; Daily et al. 1996; Eden & Ackermann 1996;

Madu 1994; Yellen 1993). Many authors have promoted the incorporation of quantitative decision making models, such as MODM and MADM in GDSS (Gray 1987; Jelassi and Beauclair 1987; Bose et al. 1997; Lewis and Keleman 1988; Pakath and Rao 1991; Jessup and Tansik 1991). GDSS of various levels of automation have been proposed to aid group decision problem solving. A number of commercial GDSSs are available which are capable of generation of alternatives, their evaluation, and selection.

Gray (1987) surveys some representative GDSS facilities, including CONSENSOR, Decision Conference, Planning laboratory, Group Decision Aid, Decision Room and Decision Conference Facility. Some of them are single purpose, some are multiple purpose; some possess the functions of interactive decision analysis, data management, graphics display and some not. These are some early-versions' GDSS. Kersten (1985) developed a GDSS called NEGO. NEGO assists decision makers in finding a compromise solution. It has been used for solving a GDM problem at the corporate level and has been utilized in management courses. Gavish et al. (1995) designed another GDSS called CM3, a distributed group decision support system. Csaki et al. (1995) presented WINGDSS, a flexible and complex GDSS for PCs in Microsoft Windows environment. Several real-life applications have been carried out with WINGDSS. Travica and Cronin (1995) designed ArgoGroup, a conceptual information system for supporting strategic decision making in groups. Group Decision Program (GDP) was implemented by Reisman et al. (1992). GDP is a topic-independent interactive videodisc-based GDSS designed to provide a computer-based, structured process to direct the pattern, timing, and content used in GDM. SCDAS (selection committee decision analysis and support) was proposed as a DSS for GDM (Lewandowski 1989). SCDAS supports a group of decision makers working together to select the best alternative from a given, finite set of alternatives. SCDAS has been implemented based on the workstation principle and can cooperate with a standard computer teleconferencing system. Another distributed group decision support system (DGDSS) was presented that facilitates the exchange of information and expertise between group members without the need for a meeting or direct communication (Jacob & Pirkul 1992). The dynamic nature of the DGDSS is achieved by viewing an organization as being composed of a networked collection of human-computer information processors. JUDGES is a descriptive GDSS for the ranking of items (Colson & Mareschal 1994). A set of tools for group decision support is embedded in this interactive software, implemented on a microcomputer. Many other GDSS can also be found in the literature.

The focus in GDSS research and in the creation of supporting software has been primarily on the group's interaction and communication (Dennis, Nunamaker & Paranka 1991) with a strong emphasis on consensus-building (Watson, DeSanctis & Poole 1988; Iz 1992a; Iz 1992b) and negotiation (Carmel, Herniter & Nunamaker 1993). However, the appropriateness of any decision method within a GDSS depends on the conditions of members, tasks and decision environment. GDSS design must take into consideration of the behavioural as well as technical issues in order to develop useful and effective systems. A technically proficient system is useless if people are unwilling to interact with it (Jelassi & Beauclair 1987). Jellassi and Beauclair suggested a framework for developing GDSS based on an integrated perspective. This framework is comprehensive and integrative as it combines the behavioural characteristics of group decision making with the technical specifications that drive GDSS. Communication channels are also an important topic in GDSS. An analysis of face to face versus computer mediated communication channels was done by Iz (1992). Experimental research on GDSS has generally focused on democratic groups whose members typically share the same objectives. In organizations, however, there are many situations where groups have a leader who has the power to override the group's recommendation, the objective of the leader may not be the same as the objective of each member, and not everyone may have the same information. Iz (1992a) reported the results of an experiment in which the groups, having a designated leader, worked on a mixed-motive task and analysed group decision outcomes and processes for groups that use a face-to-face channel of communication and those that utilize computer mediated communication. The results show two channels of communication all have advantages in GDSS. Other discussions on the introduction of GDSS in an organization (Yellen 1993), the proximity of group members (Er & Ng 1995), GDSS design (Jelassi and Beauclair, 1987, Lewis and Keleman, 1988), use of a GDSS to facilitate group consensus (Watson, DeSanctis & Poole 1988), the secretary problem in a group decision making (Salminen, Teich & Wallenius 1998), the interacting effects of GDSS and leadership (Lim, Raman & Wei 1994), the evaluating GDSS effectiveness (McCartt & Rohrbaugh 1989) have been found in literature.

However, a change has occurred in GDSS research and applications. In the 1980's, GDSS research largely was concerned with 'decision rooms' and suggestions of the impact that GDSS could have (Vogel & Nunamaker 1990). More recent research has recognized a much broader application and role for GDSS, which now are viewed as organized searching for alternatives, communication, deliberation, planning, problem solving, negotiation, consensus building, and vision sharing, as well as decision making for group members, not necessarily in the same place or at the same time.

Gray and Mandviwalla (1999) indicated in a most recent paper 'New directions for GDSS' that we have reached a point where we need to expand what we can do with GDSS. This research argued that the growth of GDSS can come in the following issues:

- increasing the capabilities available to groups so that they match all aspects of meeting;
- increasing the range of applications so that they can support more organizational decision making task;
- improving the effectiveness of group so as to achieve more productive and effective group decision making.

2.6.4 Multiple Criteria Group Decision Support Systems (MCGDSS) and Multiple Objectives Group Decision Support Systems (MOGDSS)

Literature on MCDM including MODM and MADM methodologies and applications mostly address a single decision maker. However, an organisation is frequently required to make decisions of multiple criteria or multiple objective problems in a group environment where multiple decision makers are involved. It is necessary to elicit information from decision groups and thus solve group decision problems under multiple criteria or objectives. Moving from a single decision maker to a multiple decision makers for MCDM introduces a great deal of complexity. The complexity is primarily due to the unique preference structures of the participants whose individual judgments of the 'best compromise solution' may not coincide (Iz & Jelassi 1990; Iz 1991).

A group decision and negotiation problem, in which a decision is based on multiple criteria, is called multiple criteria group decision making (MCGDM) (Iz & Gardiner 1993). For an MCGDM, particularly multiple objectives group decision making (MOGDM), each problem has multiple objectives. Each decision maker must generate relevant objectives for each problem. Each member of the group can use one or several different MODM approaches and techniques to arrive at a decision or solution. Each of the other decision makers may share some, none or all of one decision maker's objectives. The problem is no longer the design of the most preferred objective according to one individual's preference structure. The analysis must be extended to account for the conflicts and aggregation among different group members who have different objectives, goals and constraints. The group of decision makers should agree with certain rules. The group's decision is usually understood to be the reduction of different individual preferences among objectives in a given set to a single collective preference, or group preference.

In a decision group, each group member often articulates his or her preference information incompletely for objectives, goals of objectives, weights of objectives and attributes. An interactive procedure to aggregate each group member's preferences is presented by Kim & Han (1999). Since a selection is not generally made in a single step, with incomplete information, an interactive procedure is suitable for MCGDM problems. Some mathematical programming models were developed that can establish dominance relations when the preference information about utilities, goals of objectives, objective weights or attribute weights, and group member's importance weights are provided incompletely.

GDM theory, MCDM methods, and GDSS, taken together, offer great promise in supporting multiple decision makers to reach agreement on decision involving multiple criteria and objectives under a framework of DSS. The integration of these three areas produces cooperative multiple criteria group decision support systems (MCGDSS) (Iz & Gardiner 1993). MCGDSS for multiple objective decision problems are called multiple objectives group decision support systems (MOGDSS) and for multiple attribute decision problems are called multiple attribute group decision support systems (MAGDSS).

It must be noted that the meaning of MCGDSS in this thesis is to integrate MCDM and GDM methods into computerized GDSS to solve multiple criteria decision problems for a decision group (Kersten 1985). The group decision problems that cannot be modelled as MCDM problems are not addressed in this thesis.

MOGDSS is divided into two working stages:

- The first step typically consists of generating a set of candidate (efficient) solutions for consideration by the group. That is, each DM makes a decision for the MODM problem by using MCDM methods under an MCDSS.

- The second step is to find a compromise solution by facilitation mechanisms as in the discrete alternative (Iz & Gardiner 1993). That is, decision makers negotiate so as to achieve a compromise decision by using GDM method under a GDSS (Kersten 1985).

Some early researches, such as Cood and Seifoed (1978) demonstrated how to form a group consensus by combining a priority ranking scheme and a compromise programming model. Aczel (1983) illustrated the application of Analytic Hierarchy Process (AHP) in a group decision making, but didn't implemented an MCGDSS completely because of the limitation of computer technology. In the mid 1980's, NEGO that has been mentioned in section 2.6.3 was provided by Kersten (1985). Kersten's NEGO system assists decision makers in finding a compromise solution. An example was presented to illustrate how NEGO also can be used to help solve GDM problems that can be modelled as MOLP problems. Another MOGDSS was proposed by Iz and Jelassi (1990). This is an MOGDSS based on Steuer and Choo's IMOP method (Steuer & Choo 1983). The system generates efficient solutions first, then facilitates compromise through feedback on individual ordinal rankings and the group ranking based on a network model suggested by Cook and Kress (1985). Another MOGDSS using the AHP/Tchebycheff method within a hierarchical framework was developed by Iz and Krajewski (1992). After model development and group discussion on problem objectives, AHP is used to decompose and evaluate solutions to the problem at different levels. Davey and Olson (1998) studied Iz's Tchebycheff method and the Kersten's NEGO system. These systems were compared with a commercial GDSS, VisionQuest. VisionQuest does not include multiple criteria tools. The three systems were compared on the dimensions of solution quality and decision support effectiveness. However, the NEGO system was found to yield solutions with better quality measures than the solutions obtained with the AHP/Tchebycheff system. Observation of the groups using the MCDM systems indicate that both the AHP/Tchebycheff and NEGO methods can be revised to enhance their effectiveness; both MCDM techniques have potential to benefit group decision support by giving the groups a means to design better solutions.

Some recent research concentrated on the system implementation and real world applications of MCDM techniques in multiple user environments. FORMDSS is an MOGDSS for multiple resource forest management that combined MODM and GDM problem (Tecle, Shrestha & Duckstein 1998). It is a synthesis of compromise programming and cooperative game theory. This synthesis can be helpful for group decisions involving multiple criteria decision problems. Limayem and Chelbi (1997) explored the possibility of embedding decision guidance within MCDSS. This study outlined specific guidelines for the design of such guidance, and demonstrated the potential benefits of enhancing MCDM with guidance for a resource-allocation task. The findings suggest that the addition of guidance can bring significant advantages to group learning and consensus, as well as to group members' perceptions about their decision making processes and outcomes.

It is not an easy task to develop DSS for a single decision maker that meet the requirements of behavioural realism, but to develop such systems for a group is much more difficult. Often, individual decision makers are not willing to reveal their criteria or preferences to other members and they try to argue their choices by 'pseudo-criteria'. In a group, it becomes necessary to consider trade-off among preferences of group members—not only among criteria (Iz & Korhonen 1998).

2.7 Summary

This chapter draws on the existing literature on the MCDM, DSS, MCDSS, IDSS, IMCDSS, GDSS and MCGDSS. The chapter reviews the effects of the integration of DSS and MCDM, DSS and AI (particularly, knowledge-based expert system), AI and MCDM, MCDM and GDM, DSS and GDM as well as MCDM, GDM and DSS. Through a survey of MCDSS, this chapter explores whether the introduction of DSS contributes to the decision quality for an MCDM task. The chapter also evaluates whether improvements in decision quality of MCDSS typically associated with knowledge-based expert systems. The MCGDSS concepts and applications where MCDSS are removed to a group environment also are examined in this chapter. This chapter also aims to investigate the studies on integrating MCDM methods, knowledge-based systems, and GDM approaches within the DSS framework.

Results indicate that MCDSS contribute directly to decision quality of MCDM problems. Results also show that increased integration with knowledge-based systems may produce a large enhancement of MCDSS and bring significant advantages. The results also show that integrating MCDM techniques with GDSS have potential to benefit group decision support in which the decision is based on multiple criteria and objectives. In reviewing the literature, **six issues have been identified as gaps between the previous research and this study.**

1. **Most of the MCDSS are MADM type**. The primary reason for the popularity of MADSS is that MADM methods use less mathematics and more data manipulations (Minch & Sanders 1986), while MODM methods use more mathematics and more model management. Korhonen et al (1992) observed that there has been a research shift in MCDM towards providing multiple objective decision support to decision makers.

2. **All of the MCDSS are in need of the problem structuring support under an intelligent guidance (Buede 1992).** But very few studies have incorporated AI and MODSS to assess MODSS effectiveness. Particularly, the research to bring a

knowledge-based system to MODSS to guide decision makers to select a suitable method has not been found.

3. **Most of MODSS are specific methodology dependent and very specific in nature,** i.e. they deal with one specific methodology to analyse the model and obtain solution. Very few MCDSS have proper model structuring support with a MODM methodology base.

4. Despite the fact that the technology of aggregating MODM in GDSS has received some research attention, the **implementation of GDSS based on MODM (MOGDSS) with a MODM methodology base has been found rare.**

5. Most of the integration techniques of MOGDM used either some additive or multiple utility theory to find a compromise solution. Others use MADM techniques to rank the solutions generated by the specific MODM technology. New, easy to use GDM methods under multiple objectives are needed for better support. Particularly, **MOGDSS with a GDM methodology base has been rare.**

6. **There are no (to the best of researcher's knowledge) MODSS which supports the complete model management functions** (Lewandowski & Wierzbicki 1989; Lewandowski & Stanchev 1989; Korhonen, Lewandowki & Wallenius 1991) **with a set of MODM methodologies, supports suitable method selection under an intelligent guidance, and support group decision problems with a GDM methodology base under a GUI.**

Based on these six issues in previous research, this researcher addresses the use of three technologies as follows:

- MODSS---the model management of MODSS with a methodology base;
- IMODSS----knowledge-based intelligent guidance in using the MODSS resources and model building process;

- MOGDSS----solve MODM problems in a group environment through multiple communication channels.

The research reported in this thesis differs from previous research in the following aspects:

- This research integrates MODSS, IDSS and GDSS to build an Intelligent Multiple Objectives group decision support systems (IMOGDSS). This research creates both a theoretical framework and a software prototype of IMOGDSS.

- A MODM model management function with a methodology base is integrated in IMOGDSS so that decision makers can solve an MODM problem through using any method in this methodology base.

- A knowledge-based intelligent guide is integrated in this IMOGDSS so that decision maker can be recommended a best MODM method that matches the need of the specific problem.

- A GDM methodology base with five aggregation methods and two kinds of communication channels is integrated in this IMOGDSS, so that IMOGDSS can be used in both individual and group environment and can support GMD by any GDM method in this methodology base.

- This IMOGDSS prototype consists of an MODM methodology base, a knowledge base to support methodology selection, a GDM methodology base, a database and a model base and is implemented with DELPHI and CLIPS under a GUI.

- This research includes an application test, prototype evaluation, and experimental data analysis in both individual and group environment.

Chapter 3

RESEARCH METHODOLOGY AND DESIGN

3.1 Introduction

Research is defined as a "systematic investigation to establish facts or principles or to collect information on a subject" (Wilkes & Krebs 1995). A framework for research consists of a body of knowledge, a valid research methodology and understanding of the research domain (Nunamaker, Chen & Purdin 1990-1991). This chapter proposes and discusses the issues relating to the research methodology and research design of the study. A multiple-methodologies approach is applied in this study because only one methodology is not sufficient for the current complex research. System development research (Nunamaker, Chen, & Purdin, 1990-91) is the main research methodology of this multiple-methodologies approach as it is appropriate for conducting research in a information systems domain and it also uses a concept-development-impact process. The study employed system development to transform the concepts of how to integrate an intelligent technique, DSS, MODM and GDM effectively into an IMOGDSS prototype.

This chapter is organised in the following manner. Section 3.2 presents the concept of research methodology and why the multiple-methodologies approach is applied and the way system development research is selected as the main research methodology in this study. Section 3.3 describes the system development research methodology and its interaction with other research methods. Section 3.4 discusses a classical DSS development process and an adaptive development method. The object-oriented software development method and system evaluation method are presented in Section 3.5 and Section 3.6 respectively. Section 3.7 deals with the map to use these methodologies. A research design is also described in this section. Finally, the summary is discussed in Section 3.8.

3.2 Research Methodology

3.2.1 Research Methodology in the Context of Information Systems

A *research methodology* is the combination of processes, methods and tools that are used in conducting research in a research domain (Nunamaker, Chen, & Purdin 1990-91). A *research domain* is the subject matter under study in a research project. The body of knowledge includes both research domains and research methodologies. A research process comprises the understanding the research domains, asking meaningful research questions and applying valid research methodologies for these questions. Results from a research project contribute to the body of knowledge by promoting clear understanding and enhancing knowledge in a given research domain (Nunamaker, Chen, & Purdin 1990-91).

It is clear that some research domains are sufficiently narrow that they allow the use of only limited methodologies. It is also clear that some research domains are sufficiently broad that they embrace a wide range of methodologies. This is particularly true in engineering and systems where the concept at issue is likely to be viewed for its applications value rather than for its intrinsic value. This suggests that a concept with a wide-range applicability will go through a research life cycle of the form: concept-development-impact (Nunamaker, Chen & Purdin 1990-1991). A multiple-methodologies approach to information systems research is then necessary.

Researchers use one or more research methodology(ies) for conducting research to inform their readers exactly how they intend to undertake their research and how to handle data.

Galliers (1990) divided research methodologies used in information systems research into two paradigms (i.e. basic sets of beliefs to guide actions (Guba 1990)): positivist (scientific) and interpretivist.

Positivism may be described as a research philosophy assuming that the phenomena being studied have a stable reality measurable from the outside by an objective observer.

Interpretivism may be described as a research philosophy interested in human meaning in social life and in its elucidation and exposition by the researcher.

Laboratory experiments, field experiments, surveys, case studies, theorem proofs, forecasting and simulation are classified under positivism, whereas subjective/argumentative reviews, action research, descriptive/interpretive, future research and role/game playing are bound under interpretivism. System development, used as a research methodology, is provided by Nunamaker, Chen & Purdin (1990-1991). It may well be the case that systems development represents a 'super-methodology' and actually contains a hierarchy of identifiable 'sub-methodologies'. Descriptions of main research methodologies in information systems are presented in Table 3-1.

Table 3-1 Research methodologies in the context of positivist and interpretivist

Research Methodologies	Paradigm	Description
Laboratory Experiments	Positivism	The study of precise relationships between controlled variables using non-stake-holding participants solving an artificial problem. A small number of factors are allowed to vary
Field Experiments	Positivism	An experiment involving stake-holding participants dealing with a real problem. The number of examined variables will usually be small.
Case Studies	Positivism	A planned and focused investigation of hypothesised relationships in one or more organizations. A researcher is an observer.
Theory Building	Positivism	The development and testing of theorems through mathematical modelling of situations in which truth is derived based on a specified set of derivation rules.
Subjective/Argumentative Reviews	Interpretivism	An expression of the views of the research(s) based mostly on opinion and speculation derived from a range of experiences or reviews of literature
Descriptive/ Interpretive	Interpretivism	Research based on the researcher's interpretation of situations, events, phenomena, previous literature, or past developments.
System Development	--	To study an important phenomenon in areas of information systems through system building.

Sources: Adapted from Galliers (1990) and Nunamaker, Chen & Purdin (1990-1991)

3.2.2 Research Methodology in the Study

This study is about applying computer technology to multiple objective decision making in business organizations. Therefore, the necessary methods and techniques must be provided to put the technology to work in a management environment. The theories on decisions and multiple objective decision making constitute the conceptual frameworks by which decisions can be studied and understood, and by which improvements can be analysed and proposed.

The complexity of the knowledge-based and group-based multiple objective decision making context requires a wide range of methodologies. This study employed an integrated research methodology: system development research methodology, which consists of a concept-development-impact process, supported and interacted with KBDSS construction method (field experiments), software development method and software evaluation (laboratory experiments), conceptual framework (theory building) and literature review (Subjective/argumentative reviews). These research methods complement and provide valuable feedback to one another to assist the system development research. The interaction of these research methods is illustrated in Figure 3-1.

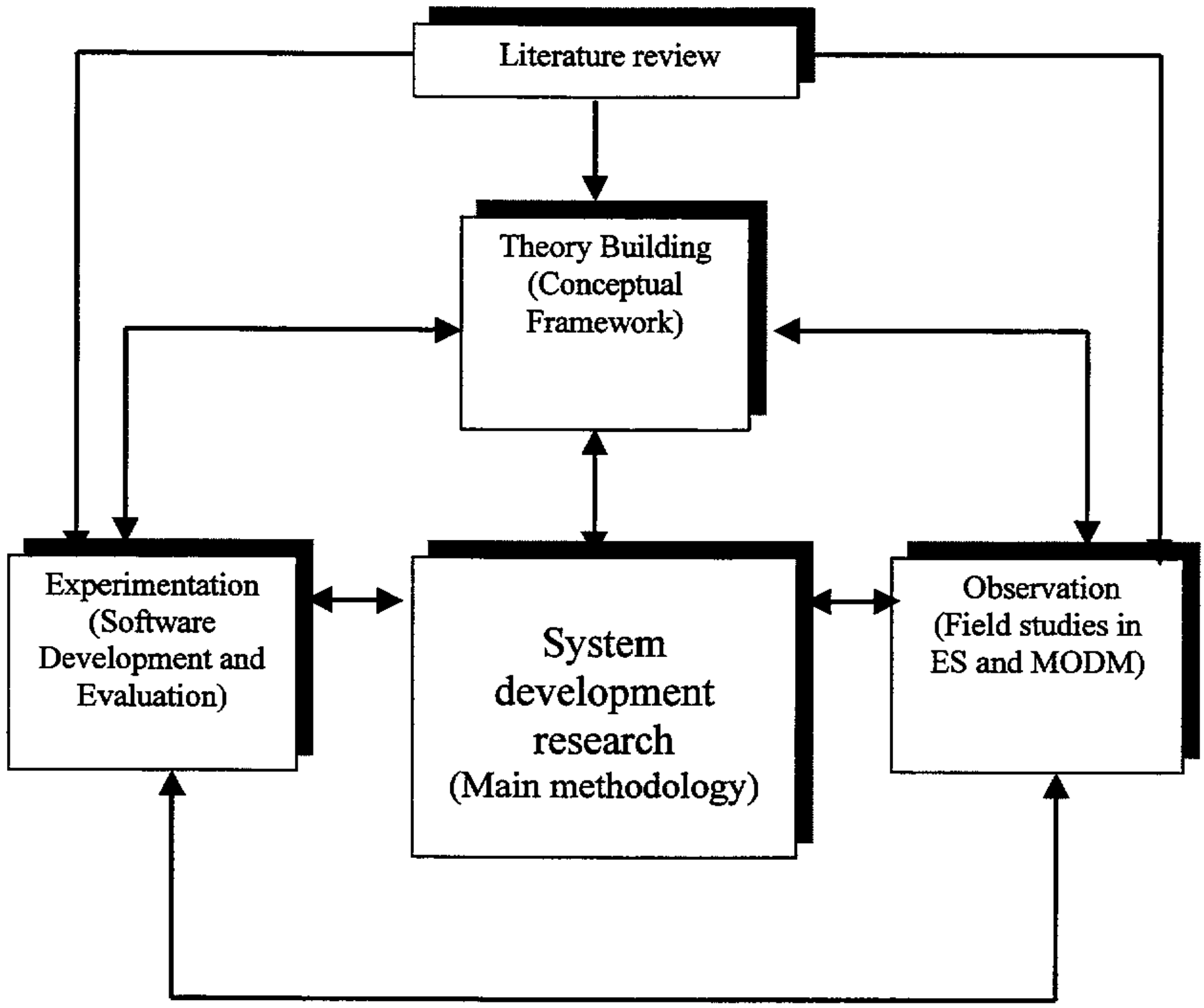


Figure 3-1 A Multiple-methodologies research methodology in this study

Figure 3-1 shows an integrated approach to the research carried out in the study, which is necessary if the research is to keep pace with technological innovation and organizational acceptance. The multi-methodological approach to this research consists of four research strategies.

Theory building: includes development of new ideas and concepts, and construction of conceptual framework, new methods, or models.

Experimentation: includes research strategies such as laboratory test.

Observation: includes research methodologies such as case studies. It may; help researchers to formulate specific hypotheses to be tested through experimentation, or to arrive at generalizations that help focus later investigations.

Systems development: consists of five stages: concept design, constructing the architecture of the system, prototyping, product development, and technology transfer.

Systems development is the hub of this research that interacts with other research methodologies to form an integrated and dynamic research program. In the current research, no one research methodology should be regarded as the pre-eminent research paradigm, because no one research methodology is sufficient by itself. In general, where multiple methodologies are applicable, they appear to be complementary, providing valuable feedback to one another. To gain a complete understanding of this complex research area, a multi-methodological approach to research is the most effective strategy.

Adaptive design and prototyping have been the key words in this research methodology because most DSS are developed by a prototyping process. Development has been focused on computer representations and program design. By means of powerful development tools, and expert system shells, rapid prototyping has been the predominant approach. It follows that evolutionary design and prototyping are important elements in a development methodology for knowledge-based DSS. They are important in reducing context complexity by performing successive re-definitions of the functionality of the computer system.

This methodology includes a descriptive aspect to develop expert problem solving models. A classical design is needed within the knowledge-based DSS (KB-DSS) conceptual framework which is presented in Section 3.4. Since knowledge-based functions will be part of this new framework, in Chapter7, this study applies a methodology, based on a cognitive approach, to the building of knowledge bases.

Testing and evaluation of knowledge bases impose problems of validation and verification. A framework for KB-DSS evaluation is also part of the multiple-methodologies approach. These problems will be dealt with in Chapter 9.

It is believed that system development and other research methodologies are complementary and that an integrated multiple-dimensional and multiple-methodological approach will generate fruitful information systems' research results (Nunamaker, Chen & Purdin 1990-1991). The premise is that research contributions can result from systems development, experimentation, observation, and performance testing of the systems under development and that all of these research approaches are needed to investigate different aspects of the research question.

3.3 System Development Research Methodology

System development is a valid research methodology and is widely used in information systems research. System development as the focal point and central methodology of the multiple-methodological approach to successful research is very important. System development research or engineering research is an artistry of design and the spirit of 'making something work'. It is a valid research technique using the proof-by-demonstration approach (Nunamaker, Chen & Purdin 1990-1991).

The idea of system development as a research methodology fits into the category of applied science and belongs to the engineering, developmental and formulative types of research (Nunamaker, Chen & Purdin 1990-1991). Applied science is the application of knowledge to solve practical problems of immediate concern or detect practical reasons. Engineering research puts emphasis on design and making something to confirm theoretical predictions. It is widely employed in software engineering areas to assist software development to analyse, design, implement, and control software projects. Development research is the systematic use of scientific knowledge to produce useful materials, devices, systems or methods, and to design and develop prototypes and processes. Formulative research (i.e. exploratory research) aims at identifying problems for more precise investigation, developing hypotheses, gaining insights and increasing familiarity with the problem area.

System development is also suitably employed for information systems research to interact with other research methodologies to form an integrated system. Generally, the advancement of information system research comes from new concepts. Then, systems have to be developed to test those concepts. Finally, the developed systems provide results for communalities or provide knowledge for future research (Kahn 1993).

The system development research methodology is employed because the system development of IMOGDSS fits well with the concept and follows the steps of system development research methodology of concept-development-impact. Initially, concepts regarding the topic were accumulated from literature reviews to identify the concrete research questions. Then, a generic conceptual framework was constructed to detect answers for those questions, guide the design of system software prototype, and conduct systematic evaluation for this software.

The principles of a system development research methodology are addressed below in the following five issues (Nunamaker, Chen & Purdin 1990-1991):

Construct a conceptual framework

This process starts from the research questions and performs a detailed literature review to investigate the system functionalities and requirement. Researcher should understand the system building processes and procedures, and study relevant disciplines for new approaches and ideas. The conceptual framework leads to theory building: different types of theory building efforts in information systems research are based on the rigidity of the 'theories': (a) declare the 'truth'; (b) formulate a concept (a framework); (c) construct a method; and (d) develop a theory.

Develop system architecture

A unique architecture design for extensibility and modularity is developed. The system architecture provides a road map for the systems building process. It puts the system components into perspective, specifies the system functionalities, and defines the structural relationships among system components. In the development type of research, researchers usually do not formulate an explicit hypothesis, but they do make assumptions about the research domain and the technical environment for developing the system.

Analyse and design the system

A research project's requirements may be driven by new functionalities envisioned by the researcher. Design, as one of the most important parts of a system development process, is rooted in engineering. It involves the understanding of the studied domain, the application of relevant technical knowledge, creation of various alternatives and evaluation of proposed alternative solutions.

Build the prototype system

Researchers in systems development often conduct their research by building a prototype system. In order to test the system in a real-world setting, however, an effort to further develop a prototype into a product is necessary. Implementation of a system is used to demonstrate the feasibility of the design and the usability of the functionalities of a system development research project. The process of implementing a working system can provide researchers with insights into the advantages and disadvantages of the concepts, the frameworks, and the chosen design alternatives. The accumulated experience and knowledge will be helpful in redesigning the system.

Experiment, observe, and evaluate the system

Once the system is built, researchers should observe the use of the system by case studies or field studies; evaluate the system' performance and usability by laboratory experiments or field experiments. The results should be interpreted based on the conceptual framework and the requirements of the system defined at the earlier stages. Development is an evolutionary process. Experiences gained from developing the system usually lead to further development of the system.

Figure 3-2 shows the five issues.

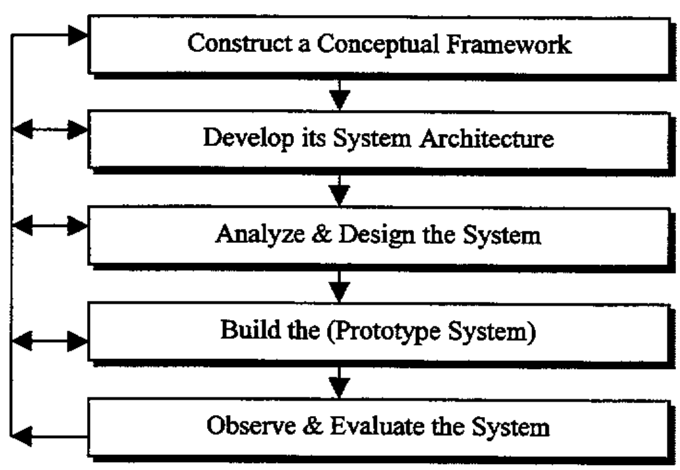


Figure 3-2 A process for systems development research (Nunamaker, Chen & Purdin 1990-1991)

3.4 DSS Development Method and Adaptive Design

3.4.1 Classical DSS Development Process

Decision support tools are becoming increasingly popular as a means of reducing the uncertainty and risk traditionally involved in decision making. The development of these tools has emerged over the past four decades following both the recognition of the importance of information in the decision-making process and the development of information technology (Adams et al. 1993). Because there are several types and categories of DSS, there is no single best approach to the construction of a DSS. There are also variations because of the differences in organizations, individual decision makers, and the DSS problem area. A comparative study of the various DSS development methodologies was compiled by Saxena (1992). He identified 32 different approaches, and discussed their major features and usability. Another survey of major DSS methodologies was conducted by Arinza (1991). He surveyed the major methodologies used for DSS development and analysed them by structure, paradigm, and orientation, and discussed their underlying assumptions.

However, because of the semi-structured or unstructured nature of problems addressed by DSS, managers' perceived needs for information will change and so the DSS must also change. Therefore most DSS are developed by a prototyping process that is different from the traditional life-cycle development process of information systems. A fundamental assumption in the traditional life-cycle approach is that the requirements can be determined prior to the start of the design and development process. However, Sprague (1980) stated that DSS designers literally "cannot get to first base" because the decision maker cannot define the functional requirements of the DSS in advance. Also, as an inherent part of the DSS design and implementation process, the user and designer will 'learn' about the decision task and environment, thereby identifying new and unanticipated functional requirements (Alavi & Napier 1984). Turban (1995) described the classical DSS development process that includes all the activities that could go into a complex DSS development. However, not all the activities are performed for every DSS. The process is summarized in Figure 3-3.

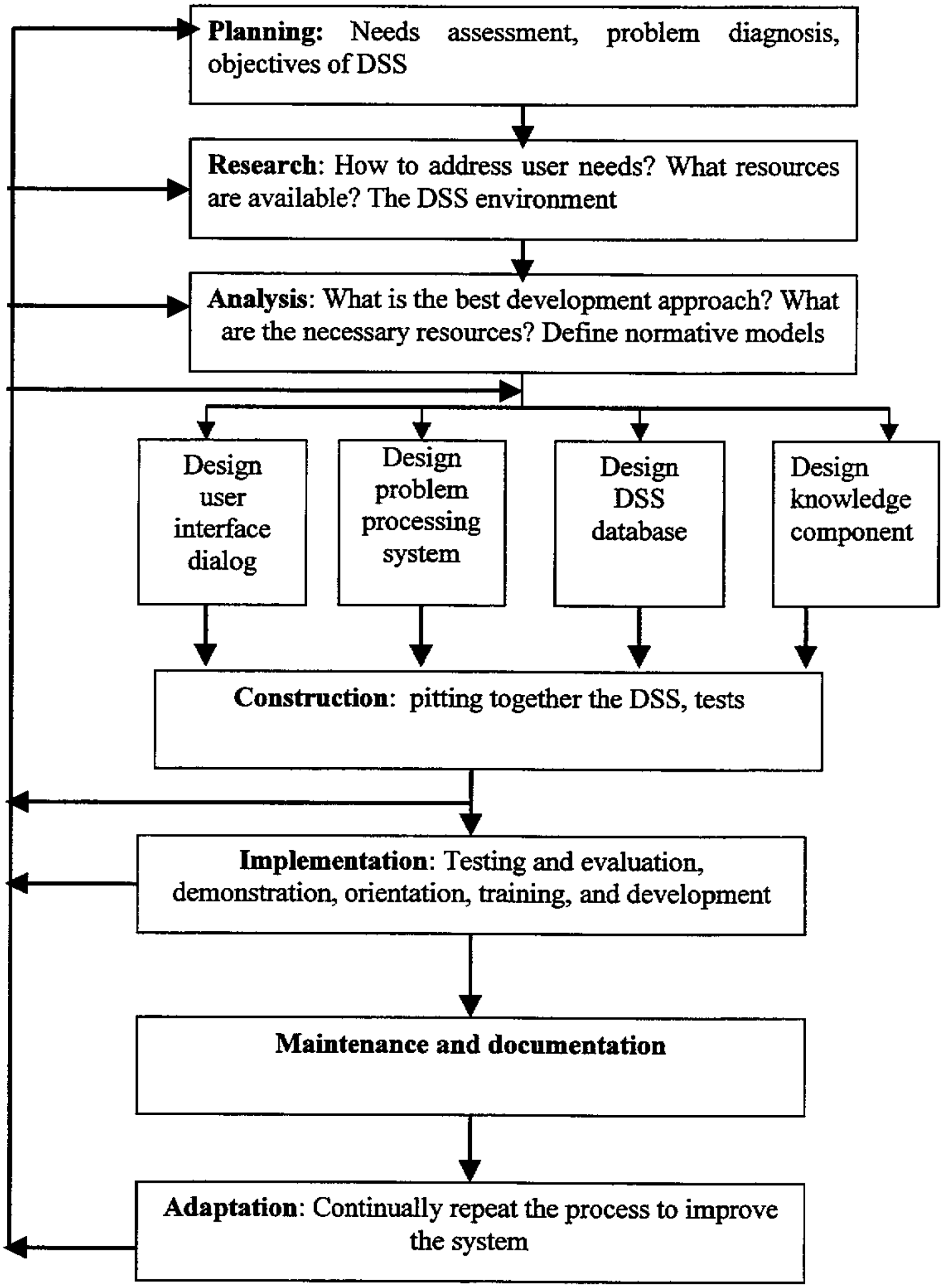


Figure 3-3 Phases in building a DSS (Turban 1995)

3.4.2 Adaptive Development Approach

In the broad sense, the DSS is an adaptive system (Sprague 1980). Sprague (1980) describes such a system as one that adapts to changes of several kinds over three time horizons. In the short run, the system allows a search for answer within a relatively narrow scope. In the intermediate time horizon, the system learns by modifying its capabilities and activities, i.e., the scope or domain changes. In the long run, the system evolves to accommodate much different behaviour styles and capabilities. Thus, the development of a DSS is actually the development and installation of this adaptive system.

Alavi & Napier (1984) focuses on the adaptive design approach and explored and discussed the conceptual issues of adaptive design. In an adaptive design approach, the four traditional system development activities (requirements analysis, design, development, and implementation) are combined into a single phase, which is iteratively repeated in a relatively short time. The process is described in the context of the framework of Keen (1976). According to this framework, the major components (elements) of adaptive design include the builder, the user (decision maker), and the technical system (DSS). During the design process, these elements interact to 'influence' each other. Hence, three adaptive links are established in this framework: the user-system, the user-builder and the builder-system. Figure 3-4 depicts these elements and their links.

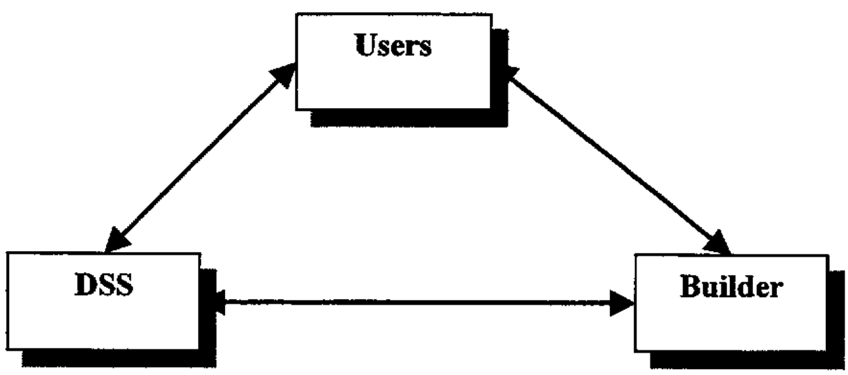


Figure 3-4 The elements of adaptive design

In this framework, the user is either the manager or individual faced with a problem or opportunity. The user is responsible for taking action and its consequences.

The DSS builder is the individual who develops the specific DSS with which the user or intermediary interacts. The builder should be knowledgeable about information systems technology and capabilities, and become familiar with the task for which the DSS is being designed. In some cases the builder may also play the role of user intermediary.

In the adaptive design framework, the technical system is the hardware/software provided to the user. A technical system is 'configured' from DSS generator and/or DSS tools. A generator is a 'package' which provides a set of capabilities to build a specific DSS quickly and easily. DSS tools are hardware and software elements applied to the development of a specific DSS or a DSS generator. Examples of DSS tools include general purpose programming languages, database management systems and expert system shell.

The user-system link deals with effect of a user's characteristics on the system utilization. The builder-system links occurs as the builder adds new capabilities and functions to the system. System evaluation and change is feasible only if system architecture is flexible; i.e., new capabilities can be added with little expenditure of time and resources. User-builder interactions involve communication and collaboration between the user and builder during the DSS development process. The iterative process includes the following four tasks (Turban 1995).

Select an important sub-problem to be built first. The builder and the user jointly identify a sub-problem for which the initial DSS is constructed. This early joint effort sets up initial working relationships between the participants and opens the lines of communication. The sub-problem should be small enough so that the nature of the problem, the need for computer-based support, and the nature of the support are clear. It should have high interest value to the decision maker even if that interest may be short-lived.

Develop a small but usable system to assist the decision maker. No major system analysis or feasibility analysis is involved. In fact, the builder and the user go through all the steps of the system development process quickly, though on a small scale. The system should, out of necessity, be simple.

Evaluate the system constantly. At the end of each cycle the systems is evaluated by the user and the builder. Evaluation is an integral part of the development process, and it is the control mechanism for the entire iterative design process. The evaluation mechanism is what keeps the cost and effort of developing a DSS consistent with its value. At the end of the evolution, a decision is made of whether to refine further, or to stop.

Refine, expand, and modify the system in cycles. Subsequent cycles expand and improve the original version of the DSS. All the analysis-design-construction-implementation-evaluation steps are repeated in each successive refinement.

This process is repeated several times until a relatively stable and comprehensive system evolves.

3.4.3 Knowledge-based System Development Process

Building a knowledge-based DSS is an incremental process where the functionality of the system evolves as experience with its use is gained. When a prototype is running, design specifications are tested, revised, and new specifications are added to accomplish needs that were not initially known. Development moves through a series of cycles before the system is finally ready for operation (Klein & Methlie 1995). Knowledge-based system development is not a linear process but follow a spiral process, as shown in Figure 3-5.

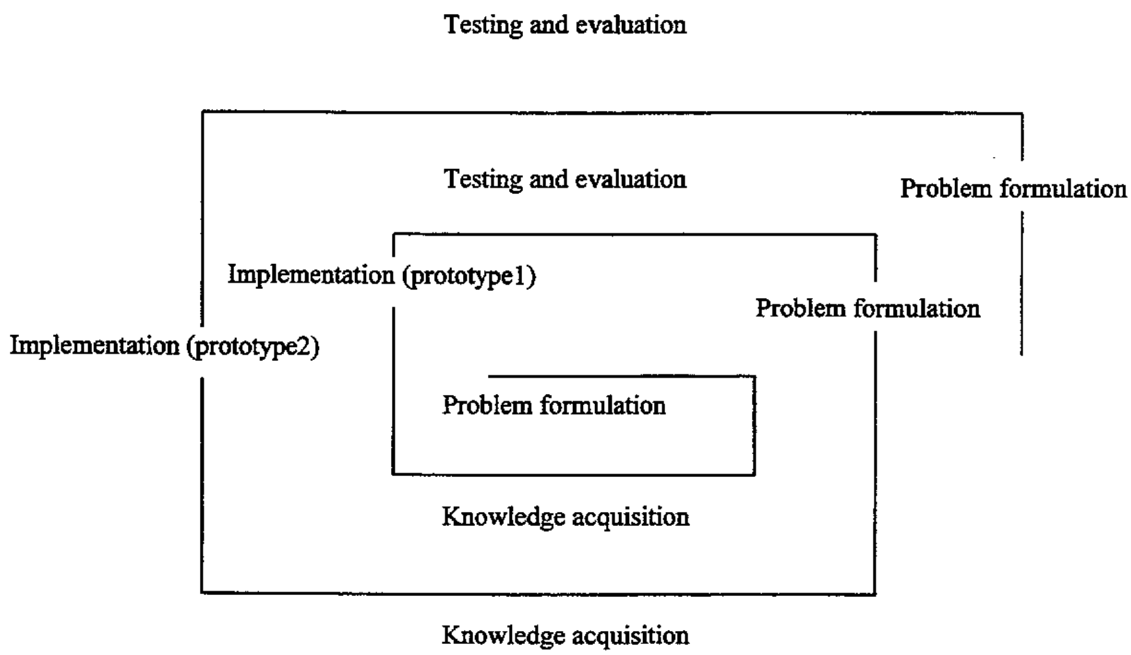


Figure 3-5 Testing and evaluation as a cyclic pattern (Klein & Methlie 1995)

3.5 Object-Oriented Software Development Method

In recent years, one approach that has shown consistent advantage when programming complex systems is object-oriented system development. Developers have found that object-oriented software development helps them deliver high-quality commercial software, and it has become the most significant trend in software development in twenty years. It is being adopted because it works (Liberty 1998).

Modular programming, structured programming, CASE, and fourth-generation languages each have attempted to address the software development problem. They have had some degree of success. Despite all the efforts to date to discover better ways to build systems, we still build systems largely by hand. Object-oriented technology can address many of the problems the software crisis presents us with.

In essence, we want code that is correct, reliable, robust, extensible, maintainable, reusable, efficient, portable and delivered on time and budget. Object-oriented techniques and methodologies serve these goals. It has been found that products built with an understanding of object-oriented analysis and design techniques tend to be more reliable, extensible and robust. They are more likely to be delivered on time, and to meet the customer's expectations and requirements.

The main benefits of object-oriented software development are improved reliability and enhanced developer productivity (Montgomery 1998). Reliability can be improved because each object is simply a 'black box' to external objects that it must communicate with. Internal data structures and methods can be refined without impacting other parts of a system. Traditional systems, on the other hand, often exhibit unanticipated side effects when a section of code is modified. Object-oriented technology helps developer deal with complexity in software development. Developer productivity can be enhanced because classes of objects can be made reusable such that each subclass or instance of an object can use the same program code for the class. Developer productivity is also enhanced due to a more natural association between system objects and real-world objects. Application development becomes shortened once the object paradigm has been learned. Object models of the world are more natural because data and programs are stored together, hierarchical model structures are possible, and successive layers can express increasing levels of detail. This all adds up to make object analysis models and software designs easier to understand, enhancing system maintenance.

Montgomery (1998) further discusses these benefits of object modelling and development:

- Faster development
- Higher quality
- Easier maintenance
- Reduced cost
- Increased scalability
- Better information structures
- Increased adaptability

Some reasons for object-oriented software development technology, such as DELPHI, are that the idea of an object that combines both static data structure and dynamic processing behaviour fits well the way we view objects in the real world. Also, object-oriented technology is an excellent way to deliver software based on reusable components. Again, objects provide solutions to business problems such that they better support adaptable and flexible business organizations. It must be noted that one important way that object-orientation will affect many business organization is in the development of event-driven, graphical user interfaces. These interfaces are driven by business events rather than by computer events.

Object-oriented languages designed for user-interface development allow user interfaces to be constructed using very few lines of program code. Because object languages help programmers to model user concepts, the user interfaces they are used to build will be more intuitive and easy to use. Objects encapsulate information about their visual forms, so programmers do not need to embed knowledge as fixed procedures.

3.6 Knowledge-based DSS Evaluation Models

3.6.1 Dynamic DSS Evaluation Model

At the end of system development research process of Figure 3-2, the system is ready to be evaluated. The evaluation phase consists of the tasks of testing and evaluation. In the testing phase, data on the system's output are collected and compared against the design specifications. During the evaluation phase, the implemented system is evaluated to see how well it meets users' needs. Technical and organisational loose ends are also identified (Turban 1996). However, Evaluation is particularly difficult with a DSS because the system is continuously being modified or expanded, and therefore does not have neatly defined completion dates or standards for comparison. The testing and evaluation usually result in changes in the design and in construction. The process is cyclical, and it repeats itself several times (Turban 1995).

The special characteristics of DSS, particularly the cyclic and evolutionary nature of its implementation, and its impact on decision making call for a special approach to its evaluation. This special approach must be distinct from the usual type of 'post implementation audit' in information systems. Evaluation is only to assess the outcome. In many cases, these cumulative evaluations suffer from 'being too little and too late.' They do not help the organization by revealing what factors led to those outcomes, and how or why they occurred.

But some evaluations provide useful information in addition to the end-product assessment. They are called formative evaluations. In the context of DSS, this formative evaluation has an added dimension, which is its dynamic and evolutionary nature.

The DSS evaluation must be an integral part of the DSS development and implementation, encompassing all phases of the DSS development process. The emphasis on decision support must be the focal point of DSS evaluation. A proposal for

DSS evaluation is shown in Figure 3-6. The evaluation process is composed of four phases: domain, design, implementation, and outcome (Athappilly 1985).

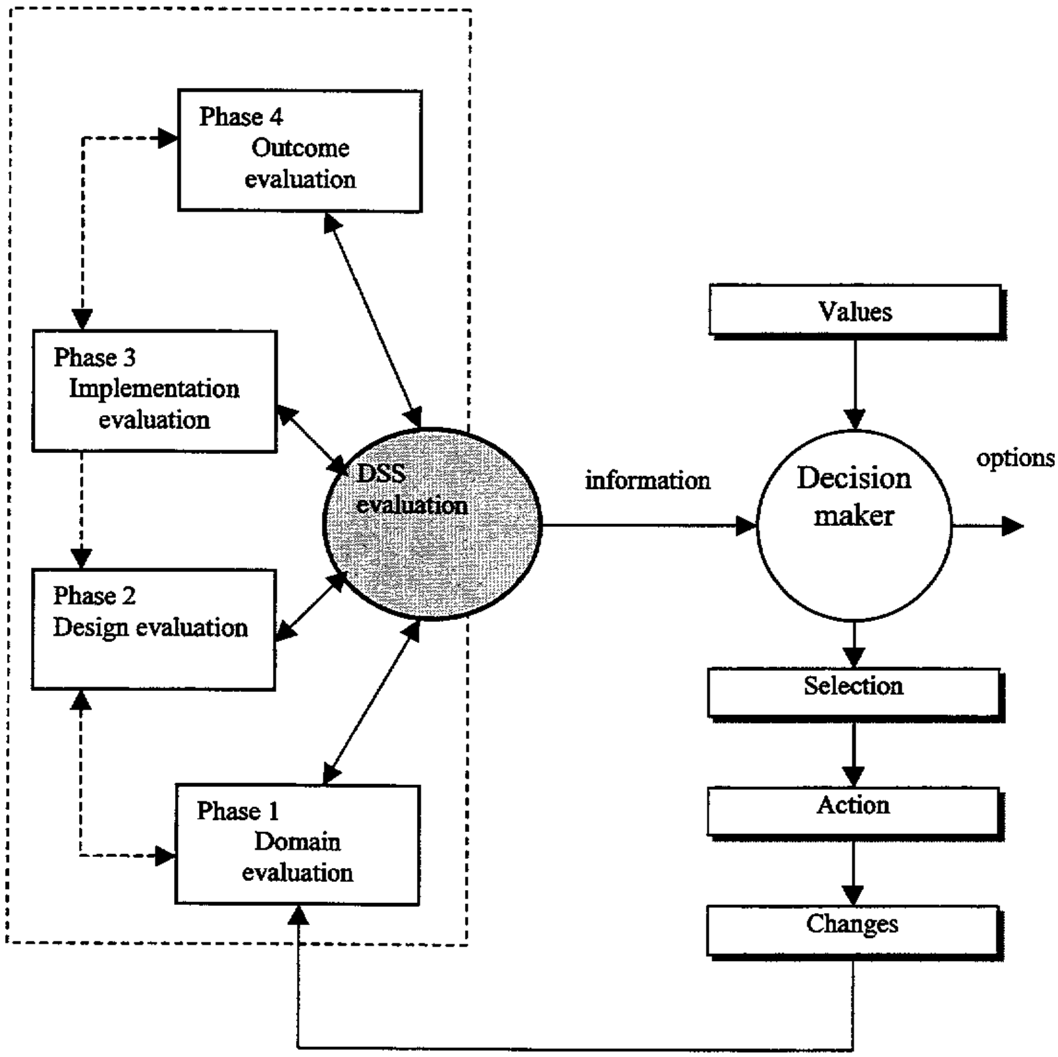


Figure 3-6 A dynamic DSS evaluation model (Athappilly 1985)

3.6.2 Knowledge-based DSS Evaluation Model

Evaluation of knowledge-based systems differs from that of conventional systems in terms of verification and validation techniques (Shahsavar et al. 1995) because of the developing knowledge-based reasoning systems introduce some new issues.

Furthermore, evaluating knowledge-based GDSS is difficult because the field is thus far comparatively unexplored.

Based on Figure 3-5, there are two aspects of concern in a revision cycle. Does the system work correctly and does it function well? Correctness can be tested; it is a question of right or wrong. Functionality must be evaluated; it is a question of good or bad.

A correct conclusion is an output expression, from the system, that can be interpreted as equivalent to an expert's conclusion in the same situation. Correctness is not only dependent on the internal logic of the program, but also on valid input specifications. Thus, correctness can only be validated against observations outside the system itself.

Reliability and consistency, on the other hand, are a question of representation and logic. A reliable system will yield the same results irrespective of the sequence in which the input data is given. A consistent system behaves in a non-contradictory way. Expert systems, although evolutionary in development, are designed for well-defined problems with predefined solution sets. This makes it possible to logically verify the knowledge base and to validate the reasoning performed by the system. Inference rules form chains of reasoning. Rules must, therefore, be tested for incorrectness, incompleteness, and inconsistencies.

In testing and evaluating a system, standards or norms are needed, with which we can compare the actual behaviour of the system to validate data, knowledge and models, and how to assess the quality of the system. Klein & Methlie (1995) provided three kinds of norms (Figure 3-7).

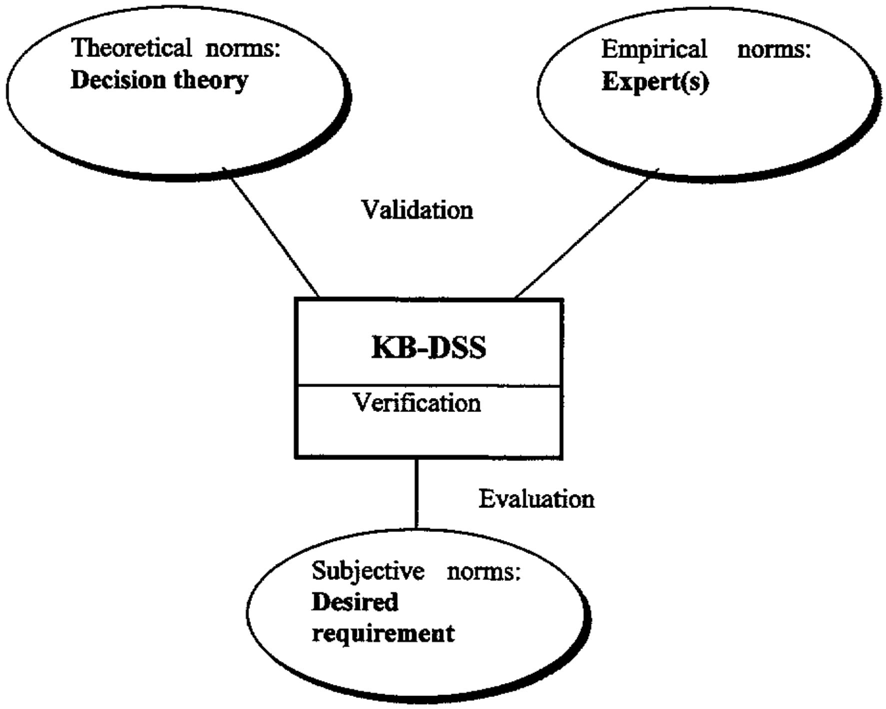


Figure 3-7 Norms as a basis for testing evaluation (Klein & Methlie 1995)

Theoretical norms such as norms prescribed by the normative decision theory. Empirical norms such as the norms prescribed by the expert which is modeled for problem solving heuristics. Subjective norms, that is, qualitative assessments of system performance, either by the users or the experts. When explicit norms are available, theoretical or empirical, the system can be validated. In the case of performance, system behaviour is evaluated.

Klein & Methlie (1995) also provided a summary of the test and evaluation framework for a KB-DSS as follows (Figure 3-8). This framework is further extended and expanded in Chapter 9.

VALIDATION
- Data input
- Knowledge base
- Reasoning
- Results

SYSTEM PERFORMANCE
- Efficiency
- Data entry
- Output format
- Hardware
- Usage
- Man – machine interface

TASK PERFORMANCE ASPECTS
- Decision making
- Decision quality
- User perception

BUSINESS OPPORTUNITIES
- Costs
- Benefits
- Values

EVOLUTIONARY ASPECTS
- Knowledge maintenance
- Response time to change demands
- Functionality of the development tool

Figure 3-8 A framework for KB-DSS evaluation (Klein & Methlie 1995)

3.7 Research Design

Once problems of a study are formulated concretely, a researcher develops a research design as a strategic plan to conduct the study. A research design provides an overall framework for system and feasible methods of data collection, handling multiple natures of data, and analysing of interpreting the data. The design is a format for detailed steps in a study to tackle previously identified research questions. Research designs depend upon statements of problems (Galliers 1990).

The research design comprises data collection and data treatment. The data for this research is both primary and secondary data. The primary data was derived from various sources: questionnaires, interview and observations. The secondary data was obtained from an in depth review of the related literature including MCDM, DSS, AI, ES, IDSS, MCDSS, GDSS and MOGDSS in order to define all the related variables and gain knowledge for research development.

The research design framework used to conduct this research was developed based on the multiple-methodologies approach. The research design consists of five stages. The details of each stage are shown in Figure 3-9.

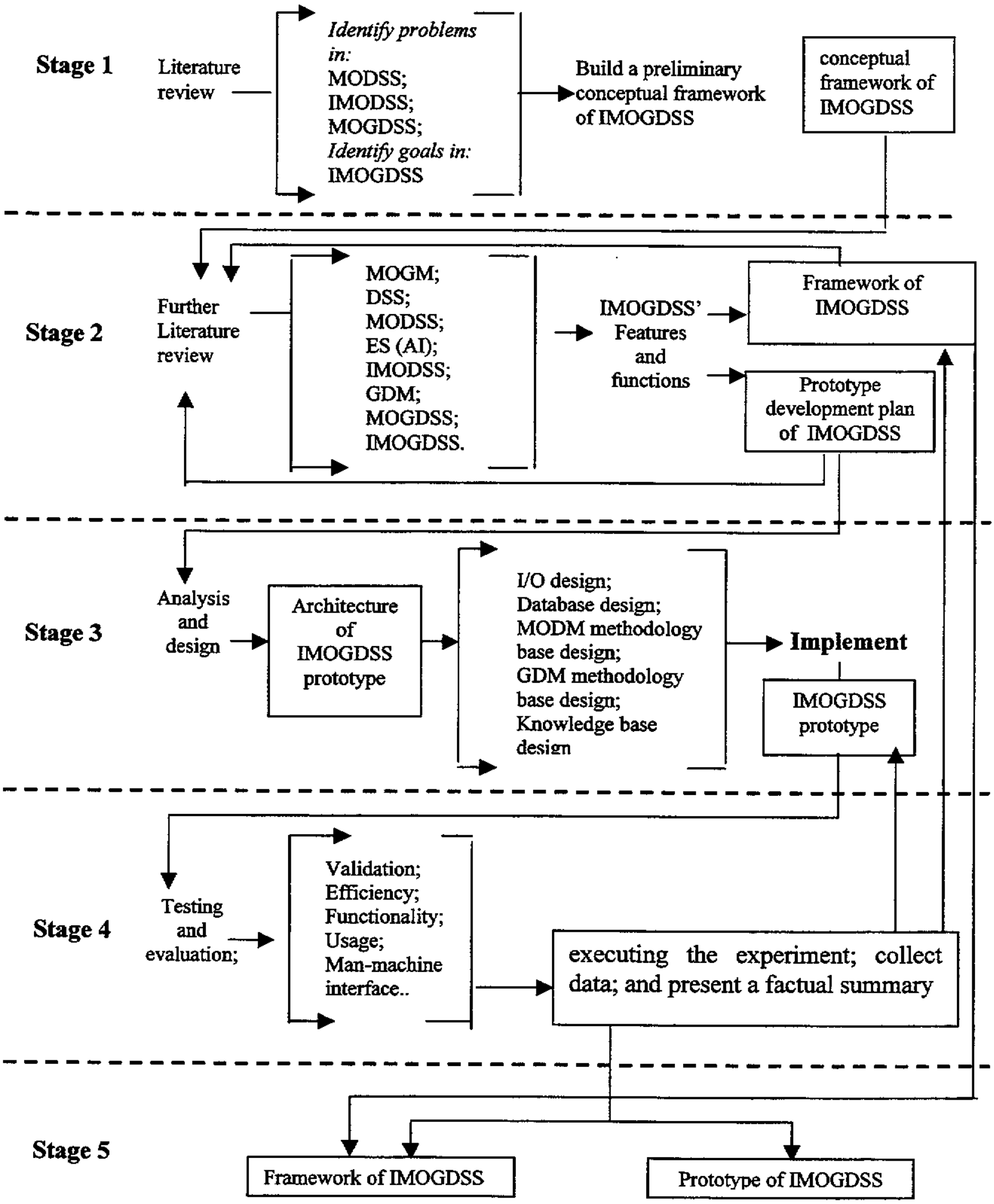


Figure 3-9 Research design

(1) Identifying Problem

At this stage, a topic for IMOGDSS is chosen. Information with respect to the topic and its problems are initially collected from literature reviews. Existing MODSS, MOGDSS, IMCDSS are reviewed for its future requirements and research questions are proposed. The obtained information is used to create the preliminary conceptual framework. The activities during this stage are as follows:

- review and analyse existing MCDSS, IMCDSS and MCGDSS respectively;
- provide a key features analysis scheme for the existing MCDSS, IMCDSS and MCGDSS;
  - identify the possible aspects of intelligence technology support in MOGDSS;
  - explore the possibility and creativeness of an integrated system: IMOGDSS;
  - propose and determine research questions;
- build a preliminary conceptual framework of IMOGDSS which combines MODM, expert system and GDM methodologies within an integrated DSS architecture.

(2) Creating a Conceptual Framework of IMOGDSS

This study will propose a new integrated framework of intelligent multiple objectives group decision making systems (IMOGDSS) which integrates an expert system as intelligent front-end, MODM methodology base, group aggregation methodology base into DSS framework. The conceptual framework is elaborated to integrate multiple system elements into one facility at the application system level. The functional integration is as a main integrated strategy in this system so that different support functions are provided as a single system. The purpose of such integration is to enhance the capabilities of the applications of IMOGDSS. Ultimately, decision makers are able to use the systems to solve this MODM problem in individual and group environment under expert system guidance. The conceptual framework, described bellow, will lead the whole research procedure.

- compare and analyse the advantages and disadvantages of existing MODSS, IMODSS and MOGDSS;
- identify and classify MODM methodologies and algorithms;
- identify the combination of expert system and MODSS;
  - identify and classify GDM methodologies and algorithms;
  - define the components of IMOGDSS and the structural relationships among these system components;
  - determine the structure of functional integration, integrated elements and integrated approach;
  - specify the system functionalities and work principle;
  - determine the integrated architecture of IMOGDSS;
  - evolve the above preliminary conceptual framework into the theoretical framework of the target system IMOGDSS;
- find some appropriate forms to express the theoretical framework;
- establish an IMOGDSS prototype development plan.

(3) Analyse, Design and Build a System Prototype of IMOGDSS

In this step we will develop a prototype for IMOGDSS where the intelligence technologies, MODM, GDM, and DSS are used in combination. Based on the conceptual framework, the functions, features and components of IMOGDSS will be implemented in this prototype. The process of analysis, design and implementation will include details of database, MODM methodology base, knowledge base, GDM methodology base and GUI. The basic framework for building a DSS shown in Figure 3-3 is expanded and applied in the prototype development. An adaptive development approach is applied in the development process to combine analysis, design and implementation into a single phase, which is iteratively repeated in a relatively short time. Object-oriented software development is used in DELPHI programming. The activities are as follows:

- analyse the user requirement for IMOGDSS prototype;
- identify an IMOGDSS prototype's characteristics in varying aspects: GUI interface, intelligence application, implementation language, software/hardware environment, communication approaches in a group to generate a 'system design diagram';
  - analyse and design the MODM methodology bases and determine which methods are used in the methodology bases;
- analyse and design the database schema and process to carry out system functions;
  - analyse and design the knowledge base as an intelligence front-end to guide DM towards the selection of the most appropriate MODM methods;
  - analyse and design the GDM methodology bases and determine which methods are used in the methodology bases;
- design a GUI for data input, solution output and menu selection;
- implement the system by using Object-oriented software development method and DELPHI.

(4) Testing and Evaluation of IMOGDSS

The system consists of knowledge base component and MODM component, and it will work at both individual and group environment. The very nature of an IMOGDSS, and the situation for which it is designed, makes it a difficult evaluation object. Since there is no predefined solution path, and the environmental context in which it is used may change from time to time, there is no single way or prescribed way of using the system. Instead, the system is a set of resources – database and model base, that are placed at the disposal of a groups of decision makers. Thus, each group decision making behaviour determines the usage of the system. The IMOGDSS may, therefore, be used in different ways by various individual decision makers and decision groups. How can the system, in these situations, be evaluated as good or bad? First of all, based on the methodology mentioned in Section 3.6, there is not one single criterion by which the integrated IMOGDSS can be evaluated. Unlike simple systems, one needs to use here a multiple-levels and multiple-factors evaluation system to test and evaluate the prototype. It is then added to prove the validation, efficiency and functionality of the IMOGDSS conceptual framework through the evaluation results from different members (different groups) of the system.

These steps will be considered in a laboratory experiment set-up as follows:

- determine a strategy for the testing process and select an appropriate set of applications;
- create a multi-level, multi-factor evaluation system to test and evaluate this system based on the framework for KB-DSS evaluation shown in Figure 3-8;
- designing experiments: understanding limitation of experiments, selecting the subjects and set up the experimental sessions;
- select application as test data;
- executing the experiment: pilot test procedure and measures extensively, running the sessions;

- collect the evaluation data and experimental materials for experiments;
- present a factual summary of the statistical analyses and draws conclusions;

(5) Report the Final Conceptual Framework and System Prototype of IMOGDSS

This stage summarises the results, explains why the results occurred and further correct the framework and prototype of IMOGDSS.

3.8 Summary

This chapter discusses the relative merits of some possible research methods and highlighted the likely application areas of each method in IMOGDSS; defines the research approach to be taken for the remainder of this work, which is outlined into multiple-methodologies approach; examines the literature pertaining to prior relative research, particularly the MODSS and MOGDSS.

These methodologies described in this chapter show how this study is processed and developed. In the following chapter, a framework of IMOGDSS is constructed.

Chapter 4

THEORITICAL FRAMEWORK OF INTELLIGENT MULTIPLE OBJECTIVES GROUP DECISION SUPPORT SYSTEMS (IMOGDSS)[2]

4.1 Introduction

Systems that provide varying degrees of decision support contain different components and are given different names including: decision support systems (DSS), multiple objectives decision support systems (MODSS), intelligent multiple objectives decision support systems (IMODSS), group decision support systems (GDSS), multiple objectives group decision support systems (MOGDSS) and intelligent multiple

[2] This Chapter is based on papers presented at several conferences:

(1) Lu, J. M.A. Quaddus and R.Williams, " Developing a knowledge-based multi-objective decision making system", *Proceedings of Hawaii International Conference on System Sciences (HICSS)*, 4-7, January, 2000.

(2) Lu, J. M.A. Quaddus and R.Williams, " A framework and prototype for intelligent multiple objective decision support system", *Proceedings of Asia Pacific Decision Sciences Institute Conference*, Shanghai, 9-12, June, 1999.

(3) Lu, J. M.A. Quaddus and R.Williams, " The design of a knowledge-based guidance system in IMODSS", *Proceedings of Australasian Conference on Information Systems*, Wellington, New Zealand, 1-3, December,1999.

objectives group decision support systems (IMOGDSS). While the names and physical characteristics of these systems may differ, their theoretical foundations and ultimate purposes are similar. And their functional frameworks and application frameworks have a 'function increasing' relationship, that is, every system brings improvements on the previous system in one or more aspects of its functional and application frameworks.

After the introduction, the chapter goes into the details of the functional frameworks and the application frameworks of MODSS, IMODSS, MOGDSS and IMOGDSS respectively. This chapter will identify the answers for the first research question of this thesis: "*What is the effective way to integrate and utilise the combined functionalities of MODM methodology, GDM approach, DSS modelling and Intelligence technology (knowledge-based Expert System) in a conceptual framework?*"

4.2 Framework of Multiple Objectives Decision Support Systems (MODSS)

4.2.1 Requirement Framework of MODSS

4.2.1.1 Multiple Objective Decision Processes

It has been discussed in Section 2.2.4 that an MODM model can be built by considering a vector of decision variables, developing the objective functions and constraints. The decision makers always attempt to maximize (or minimize) the objective functions under the constraint functions. Since this problem has rarely a unique solution, the decision makers are expected to choose a solution from among the set of efficient solutions. Mathematically, the MODM problem can be formulated as follows:

$$\left.\begin{array}{ll} \text{Maximize} & \{ f_i(\overline{x}),\ i = 1,\ldots\ldots,\ n \} \\ \text{Subject} & \text{to}: g_j(\overline{x}) \le b_j,\ j=1,\ldots.,\ m \end{array}\right\} \tag{4.1}$$

The solution of MODM problem involves the determination of one or more efficient solutions, the monitoring of solutions among multiple conflicting objectives, and the selection of the best satisfactory solution from a set of efficient solutions.

Simon (1960) divided the human decision making process into three phases: intelligence, design, and choice:

- the intelligence phase involves searching for conditions that call for a decision;
- in the design phase, possible courses of action are invented, developed and analysed; and
- the choice phase implies the selection of a course of action from the alternatives generated.

In an unstructured MODM problem, none of these three phases are structured. To solve an MODM problem, a complex set of objectives must be analysed with respect to the constraints and quantitative criteria, in order to determine the alternative that best meets the needs of a specific application (Noori 1995). The second phase of Simon's decision making process model is thus divided into two sub-phases. A multiple objective decision process that can thus be described as follows:

- the use of multiple constraints in the maximisation/minimisation process;
- the requirement to resolve conflicts and value trade-off among the objectives;
- the determination of a set of efficient solutions; and
- the process to arrive at a satisfactory solution.

We call this a *four-phases MODM process model.*

4.2.1.2 Man-Machine Interaction Types in Multiple Objective Problem Solving

The man-machine cooperation in multiple objective problem solving is very important. The interaction activity will be throughout the whole multiple objective decision problem solving process. There are different types of interactions among the MODM methods and any specific method takes one or more of these types (Quaddus 1997).

- The first type is that, an interaction is performed before the solution process even starts. In this type an explicit preference function of the decision makers are needed.
- In the second type, the preference information of the decision makers are needed during the solution process. Here the decision makers are required to provide on line preference information, but no explicit preference function is needed. This type of approach is widely known as the interactive approach.
- The third type of approach requires preference information after a set of candidate solutions has been generated. In this case the decision makers are simply required to choose the most satisfactory solution from the set.

We call the three types of interaction as *pre-interaction, pro-interaction* and *post-interaction* respectively. The three types of interactions are specified in Figure 4-1.

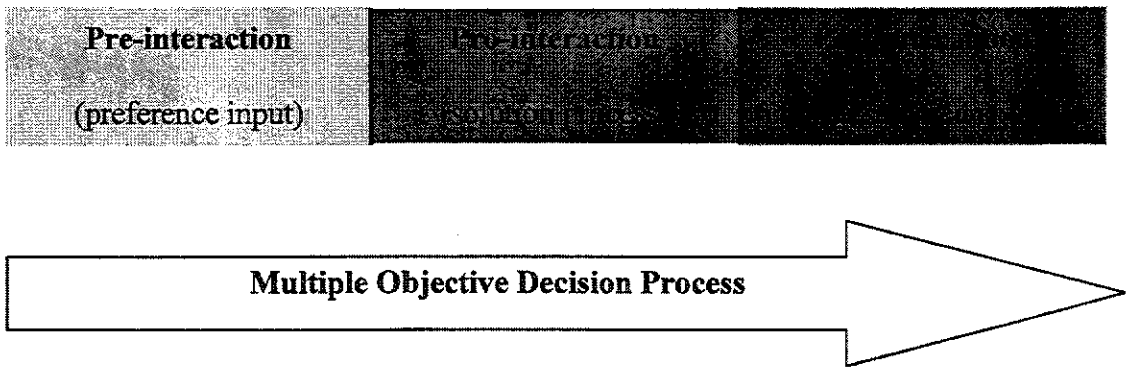


Figure 4-1 Three types of interactions

4.2.1.3 Objectives of MODSS

An MODSS can be defined as a computer program that provides information (solutions) by means of multiple objective decision models and methods, and accesses a database for objective functions, constraint functions and decision variables in order to support decision makers in making decisions effectively in multiple objective decision tasks. The objectives of an MODSS are to improve a multiple objective decision making by better understanding and preparation of the decision making process; to help decision makers make decision analysis; to support a satisfactory solution's formation and selection (Noori 1995). However, an MODSS must guide and support the decision makers to complete the four-phases decision process discussed in 4.2.1.

4.2.1.4 Characteristics of MODSS

Decision situations under multiple objectives are inherently ill-structured. This has been mentioned in Chapter 2. In an MODM process, the implicit or explicit trade-off among the criteria are called for and the extensive human intervention are required to arrive at the 'best compromise' or 'best satisfactory' solution. Usually, an MODSS first generates a small set of efficient solutions to facilitate a decision maker exploring and compromising through interaction with the system. A final solution is obtained based on the feedback of the decision maker's individual preferences for a satisfactory solution.

An MODM problem solving task is usually broken down into a series of subtasks:

- MODM problem recognition;
- MODM problem diagnosis and structuring;
- MODM problem modelling and methodology selection;
- a set of efficient solutions generation as alternatives;
- monitoring or modifying of the solutions;
- choosing between efficient solutions; and
- reporting a final solution.

Therefore, an MODSS has to posses the following characteristics in order to support the above MODM subtasks (Klein & Methlie 1995):

- Information access---to able to access objective functions, constraint functions through a database and accept the decision makers' preferences for the goals, weights and priorities with respect to the conflicting objectives as well as the efficient solutions.

- Communication---to provide a way to allow decision makers to monitor goals, weights and priorities of objectives and explore more satisfactory efficient solutions through interaction with the system.

- Decision situation structuring and modelling---to able to define and structure a multiple objective decision problem, and generate users' decision making models based on their data input.

- Computation of decision criteria---to able to search for a solution based on MODM formalization, computation and data manipulation, and form a set of efficient solutions by the use of multiple constraints in the maximisation process.

- Decision analysis---to able to resolve conflicts and value trade-off among the objectives, and produce an ideal solution and a set of efficient solutions, then allow decision makers to accept or reject every solution and help them arrive at a satisfactory solution (based on the decision maker's judgments).

- Reporting---to report the final satisfactory solution and related information.

Each characteristic of MODSS supports the implementation of one or several MODM subtasks and each MODM subtask can be supported by one or several characteristics of MODSS. Figure 4-2 shows these subtasks of MODM and the support from MODSS' characteristics.

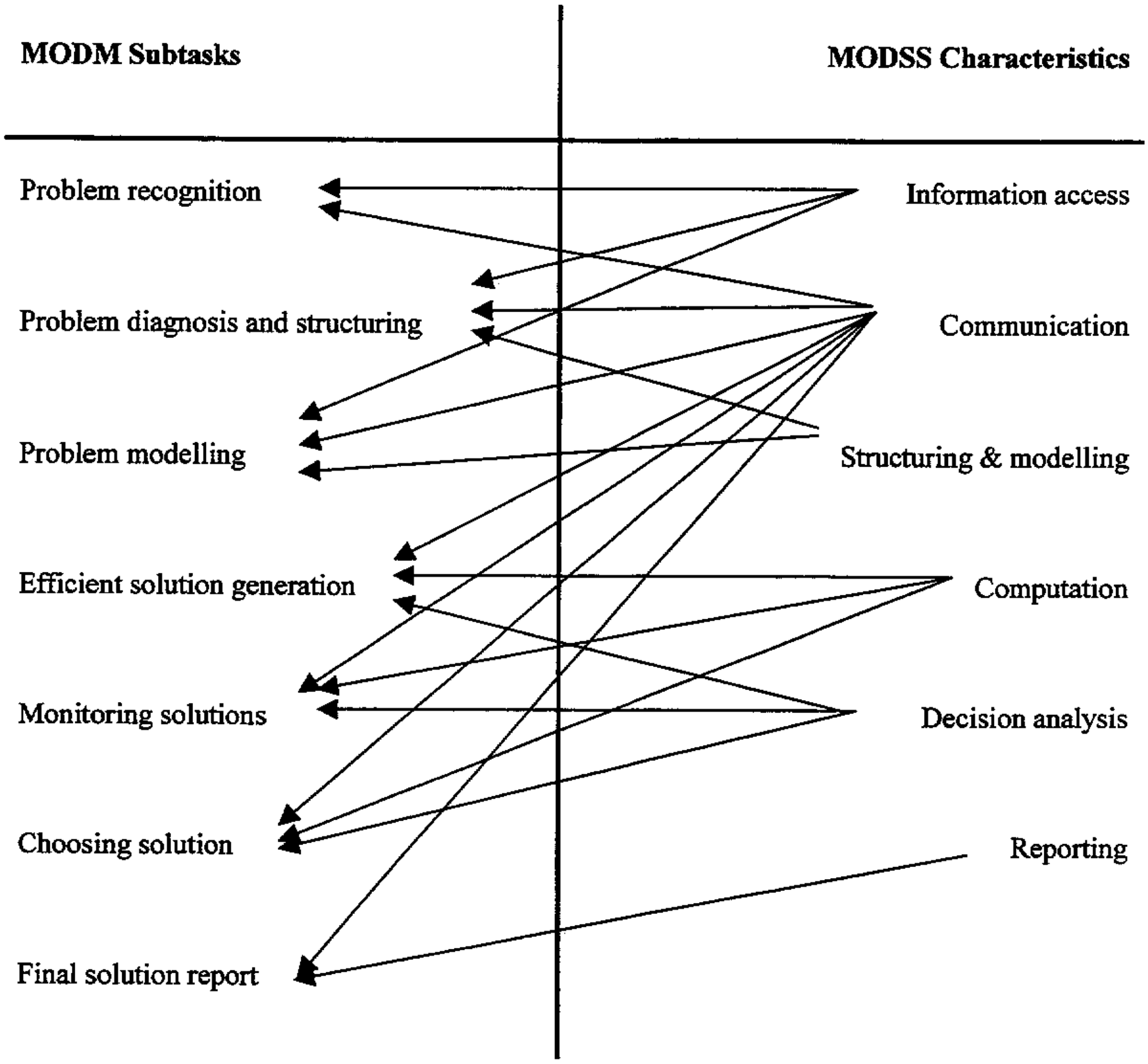


Figure 4-2 A requirement framework of MODSS

4.2.2 Functional Framework of MODSS

Given the above characteristics about an MODSS that we wish to reach, it is well accepted that certain requirements and functions have to be fulfilled by MODSS software. The functional components that an MODSS should have can be derived from the main steps in the decision making process and the requirements for MODSS software. Generally, an MODSS consists of the following functional components.

- model management component

The model management component defines, develops and maintains the decision models needed to compute the decision criteria and efficient solutions. An MODSS may contain several user MODM models. These user models may be defined and developed over a long period of time by different people. The component is able to manage these models and each model may be used and modified any time. This function implies the existence of a model base.

- data management component

The data management component will define and manage the collections of entities and their attributes represented as data structures in each MODM problem as well as the solution for these problems. In most MODM problems the first step towards a satisfactory solution is ready access to relevant data since the decision problem recognition is through the data access. The database access includes objective functions, constraint functions and decision variables. It is important to be able to have access not only to the values of the coefficients of objective functions and constraint functions but also the meaning of these objectives and constraints.

- methodology management component

It is found that some methods are more suitable and efficient than others in the solution of a particular decision problem of particular decision makers. Hence, an MODSS tool should preferably contain a sufficient number of methods for the decision makers. This function implies the existence of a methodology base and a methodology base management system in order to access specific algorithms used to solve various MODM problems. Decision makers can select the most suitable method for solving their decision problem. To harness the potential of these methods effectively, the system must possess two important characteristics. Firstly, it should be flexible enough so that new or revised methods could be introduced when the need arises. Secondly, it should be capable of guiding users possessing various levels of expertise on the usage of these MODM methods.

- input management component

This component will allow decision makers to enter data including coefficient and their meaning of objective functions and constraints, define and select events that they want the system to accomplish.

- output management component

This component will define reports to present the various kinds of information used during the search for a solution in a way that is adapted to the problem and decision makers. This component also defines how the system resources are displayed and gives a way for decision makers to interact with their applications.

- problem solving component

This is a synthetical function component and it will read users' data and requirements, realize this problem's modelling, select a suitable MODM method and run an algorithm to get a solution through an interaction process, and finally transfer the solution to reporting.

The input component, output component and problem solving component imply the existence of an interface to let a decision maker define or select what she or he wants the system to accomplish and give feedback for what the system has completed. It also implies a high level of interaction so as to be able to see clearly the consequences of changes in the goals, weights and priorities of all objective functions. Figure 4-3 shows the conceptual functional framework of MODSS. This is a basic functional framework. It will be extended to produce other functional frameworks in the following sections.

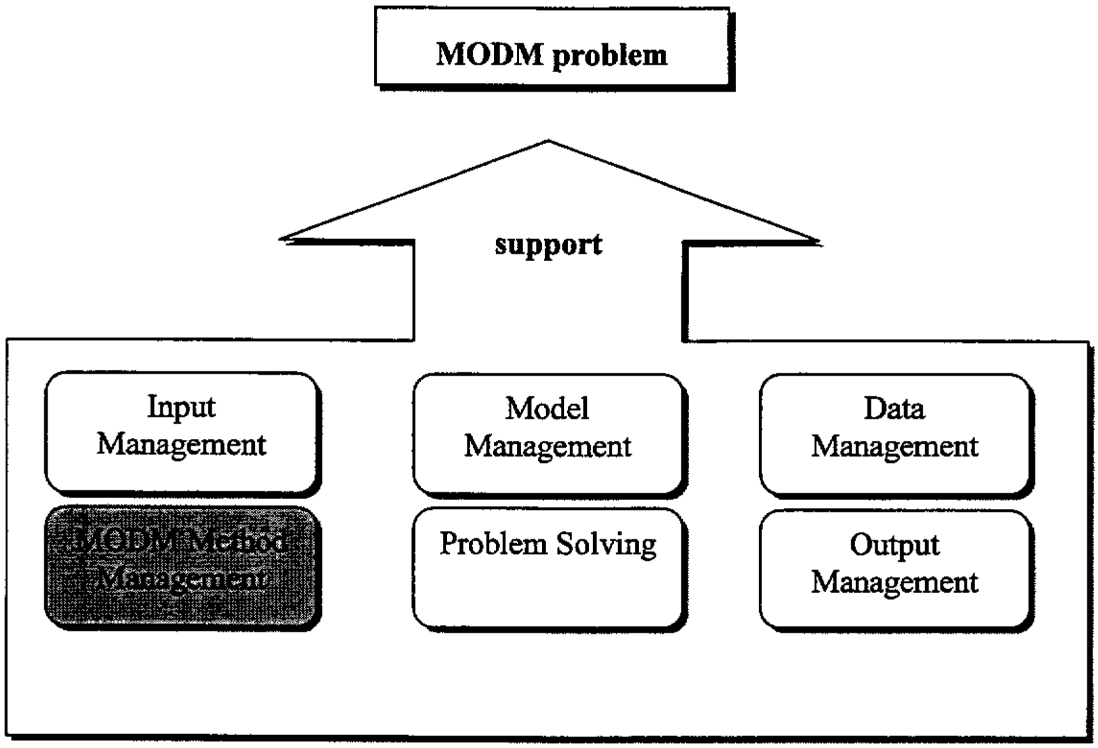


Figure 4-3 A functional framework of MODSS

4.2.3 An Application Framework of MODSS

Early MODSS (and DSS family) were computer systems that utilized decision rules and an MODM model coupled with databases to help decision makers solve MODM problems (Bonczek, Holsapple & Whinston 1981; Sprague 1980). They were only used as an interactive computer-based aid designed to assist managers in complex MODM tasks requiring human judgment. Many of these MODSSs had an intrinsic capability to support ad hoc data analysis and reduction in the multiple objective decision modelling activities (Radermacher 1994), but where the systems only do what the users explicitly direct them to do. For example, they answer what-if kinds of questions, and support a narrow range of decision making tasks.

With the new generation of DSS, a new definition for MODSS that provides active support to the decision maker has been proposed (Rao, Sridhar & Narain 1994). An active MODSS operates almost independent of explicit direction from the users to provide support that the user finds helpful. A computer maintains one or more updated models of the user problem working process, provides a set of MODM methodologies, and complements the user's work in a number of forms, such as suggesting alternate actions.

The new generation of MODSS has two independent processes, one user directed and the other computer directed. The user initiates the problem solving process and subsequently the computer may activate some processes, which would operate without explicit direction from the user. The result of the computer directed process can be used by the user in arriving at a solution (Rao, Sridhar & Narain 1994). The two independent processes are shown in Figure 4-4.

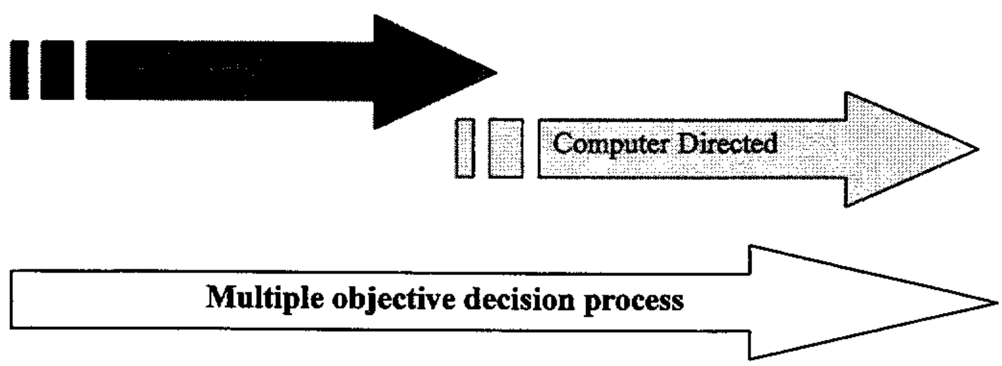


Figure 4-4 A user directed process and system directed process

Based on this new concept and the DSS principles discussed in Section 2.3.3, an MODSS requires several physical components for an efficient and effective functional framework. These physical components include:

- An *interface* for decision makers to input their decision problems, interact with the system and obtain the solutions. All MODSS applications mix different kinds of interfaces: windows, menus, dialog boxes, icons, etc. It is essential to use a GUI and the graphical display function which will provide an assistance function to MODM models and solutions to aid the decision maker to understand, analyse and choose.

- An *MODM methodology base* for computing, exploring and analysing the solutions. The methods in the methodology base are developed as independent executables to facilitate the flexibility required of the system. These methods share similar data acquisition routines and these routines are developed as independent modules so that data acquired could be accessed by all the methods.

- A *data base* and a *database management system* to store and maintain data resource.

- A *model base* for users application models. These models in the model base are linked with the database. The model management system can draw parameters, coefficients, and variables from the database and enter results of the model's computation in the database. These results can then be used by other models later in the decision making process.

These *physical components* are used to support a set of capabilities of an MODSS through multiple purpose interactive processes between decision makers and the system, and information exchange among the three physical components. The conceptual functional components shown in Figure 4-3 are converted into these physical components of an application framework of MODSS described in Figure 4-5.

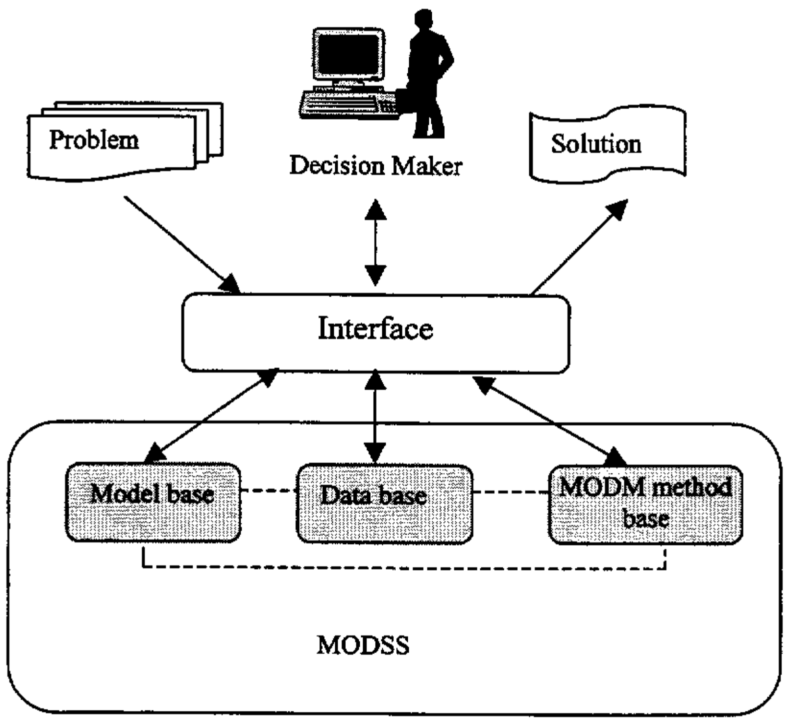


Figure 4-5 An application framework of MODSS

4.2.4 Development Framework of MODSS

There are many differences between the traditional information system and a DSS (and an MODSS) including:

- a different user relationship;
- different user expectations; and
- different data processing expectations (Leighton 1981).

DSS, including MODSS, cannot be designed using the life cycle approach typically applied in the design of transaction processing systems (Sprague 1980). Now, many DSS development approaches have been proposed, such as an adaptive design approach which facilitate DSS design and development where the systems development phases of requirements analysis, design, development, and implementation are integrated into a single phase (Alavi & Napier 1984). The Object-oriented approach has been used as a main development approach for DSS (Power & Saarenmaa 1995; Wolf 1994; Muhanna 1993). Keen (1980) argued that DSS applications tend to rely heavily on an iterative development process that involves multiple executions of a combined analysis-design-implementation effort.

However, a development process for a DSS (tool) includes the same basic steps that are required for most applications—idea generation and approval, information requirements determination, design, development, and testing (Sprague & Hugh 1995). Based on the application framework of MODSS discussed in Section 4.2.3, we further divide the step 'development' into a set of sub-steps: interface development, database development, model base development and MODM methodology base development. Therefore eight major steps involved in the development process of an MODSS are shown in Figure 4-6. The related decision aiding techniques and related fields' knowledge applied during the development process are also listed.

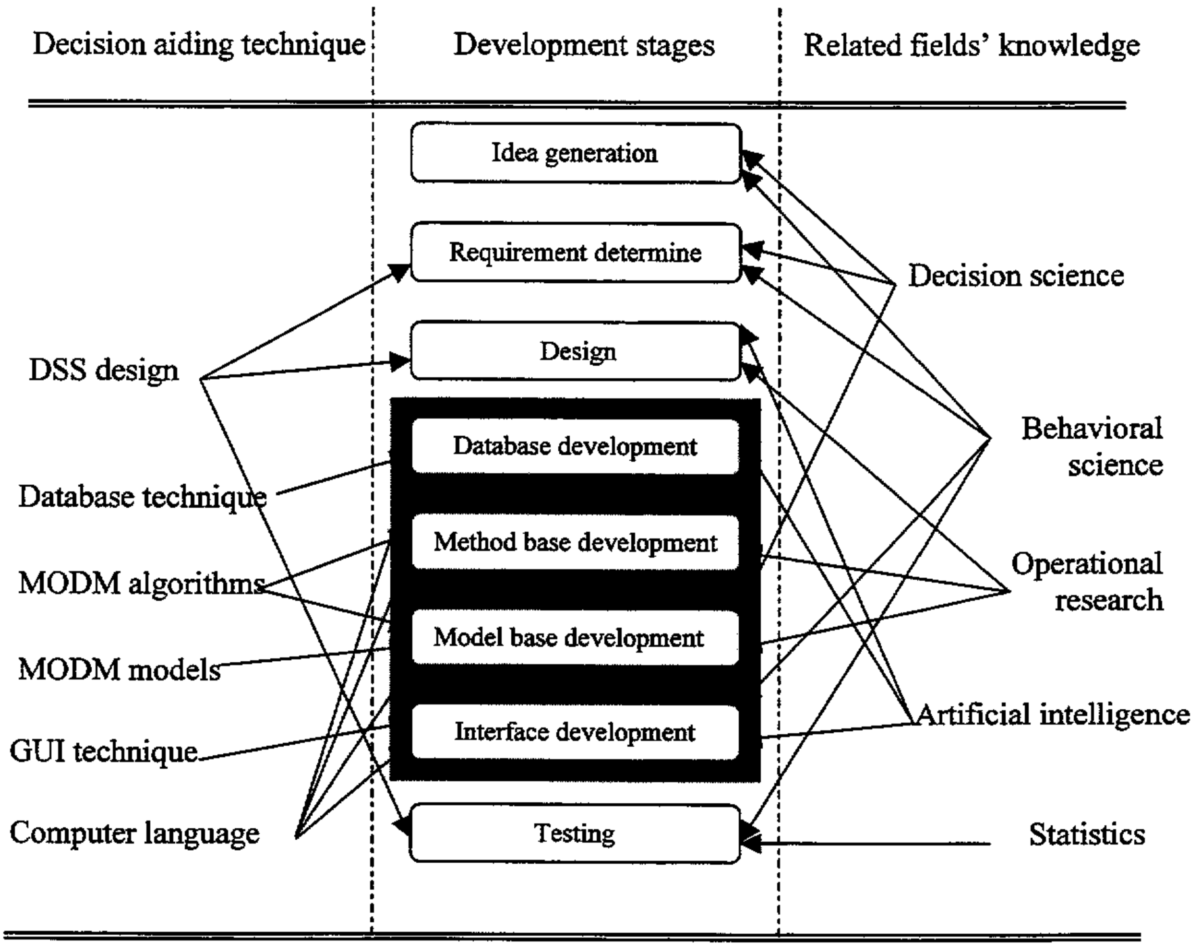


Figure 4-6 A development framework of MODSS

4.3 Framework of Knowledge-based Intelligent MODSS (IMODSS)

4.3.1 Requirement for Knowledge-based Intelligent Guide in MODSS

Integrating a knowledge-based intelligent component in the MODSS conceptual framework will considerably increase the expertise embedded in the MODSS and will improve the capacity for users to enhance this expertise (Klein & Methlie 1995; Reitman 1990; Budgen & Marashi 1988). Some researchers have emphasized the need for research in the intelligent MCDSS (Limayem & Chelbi 1997; Iz & Gardiner 1993), explored the possibility of embedding intelligent decision guidance within MCDSS (Poh 1998), and provided a conceptual framework for a knowledge-based DSS (KBDSS), and defined the basic functions of KBDSS applications (Klein & Methlie 1995) (related details have been mentioned in Section 2.5).

An MODSS application requires integrating a knowledge-based intelligent component because users in general,

- lack the expertise and experience to understand specific features of an available MODM method;
- lack the experience to systematically analyse and decompose complex problems into subtasks;
- lack the ability to match the most appropriate MODM techniques with current problem subtasks;
- lack the ability to codify human expertise permanently;
- lack the opportunity to explore 'what if...' situations with different types of inputs;
- lack the ability to handle uncertainty when data are incomplete; and

- lack the ability to solve problems that have extremely large sets of possible solutions to explore.

4.3.2 A Framework for Knowledge-based Support in MODSS

Once knowledge is embedded into an MODSS, the functions of the knowledge-based IMODSS will be enhanced to provide 'intelligent assistance' to the decision maker with the following functions during its problem solving task. The architecture for knowledge-based support in an MODSS is shown in Figure 4-7.

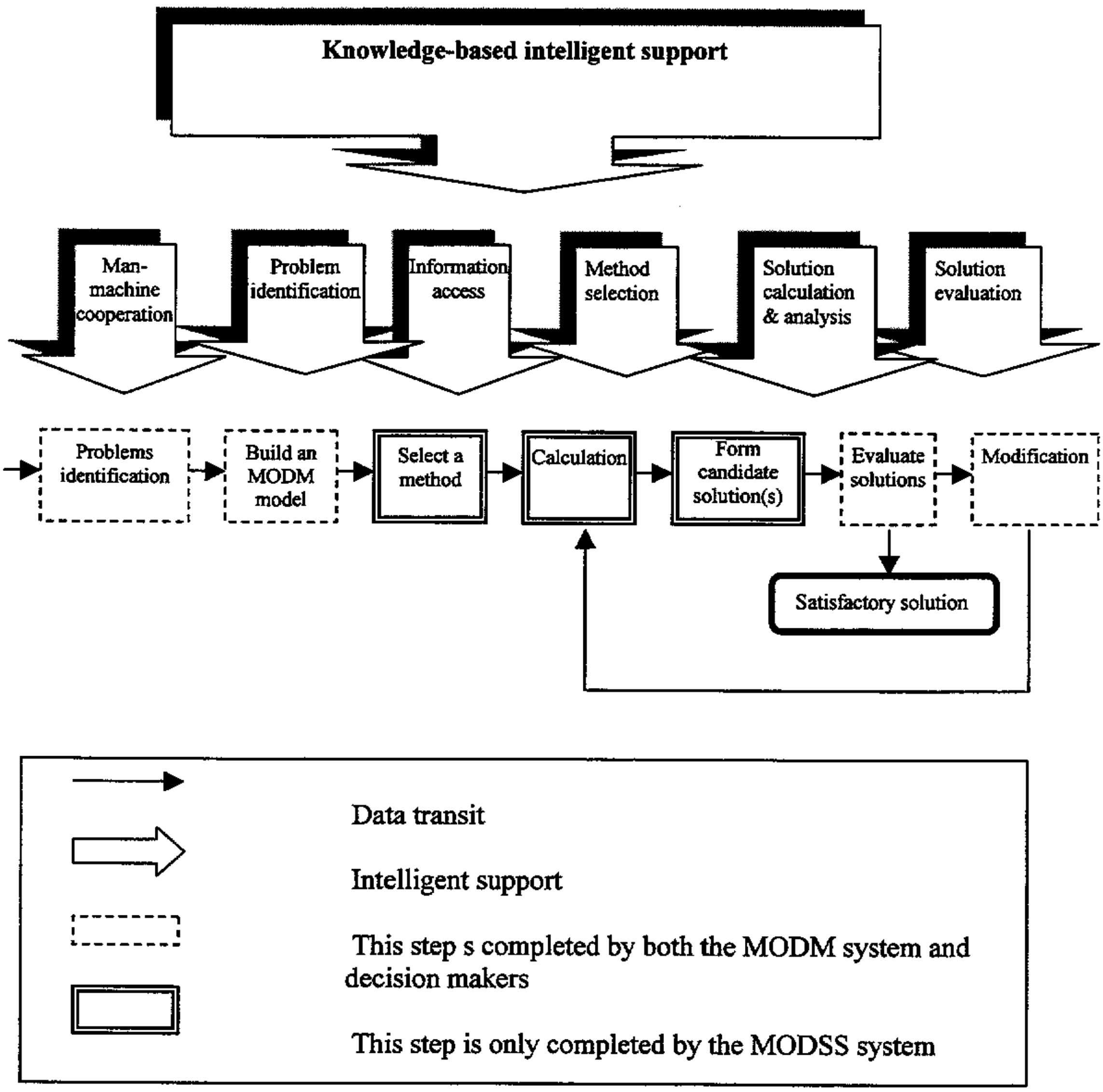


Figure 4-7 An architecture for knowledge-based support in IMODSS

Each of the knowledge-based intelligent support in Figure 4-7 is all elaborated below.

4.3.2.1 Supporting the Man-Machine Interaction

Most MODM methods involve a man-machine interactive process. There are three main types of man-machine interaction: pre-interaction, pro-interaction and post-interaction as mentioned in Section 4.2.1. Whatever the type of interaction, a knowledge-based guidance plays an important role by providing expertise when human expertise is not available, assisting the decision makers in choosing a suitable analysis approach and generally assisting in complex situations.

An IMODSS provides an intelligent interface to support the man-machine interaction during the problem-solving process. The interface of an MODSS defines how the systems resources are used and gives a way for decision makers to interact with their applications. The interface of an MODSS application is a key element for its success because this is the user's view of the system and the unique way to input objective functions and constraints conditions, to analyse solutions and to receive a satisfactory solution. It is very important in an MODSS because the decision situation under multiple objectives is inherently unstructured. This object oriented graphical interface is usually made of a hierarchy of menus and icons giving access to resources that enable the decision maker to select various functions in the decision problem solving process.

Choosing and displaying information about decision variables, objective functions, constraint functions, goals, ideal solutions and satisfactory solutions needed for MODM problem solving, are the most usual functions provided by the interface of an MODSS. However, for more complex interactive actions, e.g., to make a solution analysis by relaxation in the STEM method (Benayoun et al. 1971), knowledge-based guidance will be needed to support the assistance function to help the decision maker. The more important roles of the interface are to help users select between alternatives and display/store a report. The application interface is a layer between the decision maker and the resources of the application. Usually, an MODSS application requests a wide range of techniques for the user interaction. Selection of options in a menu or icons with guidance is fine for calling a resource such as a data file, a database table and an MODM methodology.

4.3.2.2 Supporting Identification of the Application Problem

In many real situations, the first step is not even to assess the objectives of an MODM problem but to understand what an objective is, i.e. problem identification and recognition. For the support of the problem identification phase, the decision makers should be guided by the system to determine which questions (decision problems) are to be answered, define the hypotheses to be tested and the effects to be estimated.

The first aspect for supporting problem identification is to show data and appropriate messages. Since the essence of problem recognition lies in the ability to detect differences between the present state and expected states, for MODSS, these differences are reflected in differences between goals and efficient objective values. The capacity to present information in windows with colour graphics is a key element of a good display of information for identification of application problems. An example of the use of an expert system in order to build an alternative in MADSS is presented in Bois et al. 1989, where a frame-based system determines the possible pathology of a patient.

The alerting function of an MODSS is also a very important function with which to support problem recognition. The idea is that the system should be able to set conditions (predicates and rules) based on the decision makers application which, if they become true, will trigger the display of a report, or send a message, or, eventually, trigger some other action such as the computation of a model. This function is very useful when the decision maker is confronted with a large amount of data that is evolving very fast with time (Klein, 1995). An 'intelligent' alerting system must be defined. The knowledge-based rectification guidance feature is one of the ways of alerting the user.

4.3.2.3 Supporting Access to Information During Decision Problem Solving

Easy access to the decision information and the system resources are the first support a decision maker requests from an MODSS. The resources of the MODSS include the database, the users' decision models, the MODM methodology base, available displays and reports produced. An IMODSS should provide information about the resources of the system in terms of data useful for solving the MODM problem. The related information includes: the meaning of models and methods that are used in studying the problem; the meaning of variables used in MODM models, the conditions of use, and the underlying hypothesis of MODM models; the use of available reports; the ways to select a satisfactory solution, solutions analysis strategies, menu options and help. A classification for accessing information in MODSS is shown in Table 4-1. A knowledge base can be defined to help the decision maker using the application to select the right resources in terms of decision models, reports, databases and method bases.

Table 4-1 A classification of information access

	Information source		**Acquisition method**	**Type**	**Information example**
		System information	System message & variables	Text	Error message box
1	**Implicit**	Guidance information	Instructions	Text & variables	The number of solutions & the caption of forms
		Statement information	Implicit trade offs	Variables	A cycle No in a solution process
		Interactive	Interact with applications	Model	Make a relaxation of a satisfactory objective
2	**Explicit**	Knowledge support	On-line reports	Data	Questions for selecting a suitable method
		Record	Users database & model base	Data	All of the above

4.3.2.4 Supporting the MODM Methodology Selection

It is important that an MODSS contains a sufficient number of methodologies in its methodology base for the decision makers' use during the solution process. However, it is difficult to select the most suitable method from such a methodology base because of the dearth of expertise and experience needed to understand the specific features of the available MODM methods, as well as the ability to match an MODM model with current decision needs.

It has been frequently reported that decision makers are often troubled by the choices of suitable methodologies according to their various applications. Usually only experts in the field are able to take full advantage of an MODSS. This is because sophisticated analytical skills on the part of the users are required to identify the problems and to match each problem with appropriate MODM methods. At the same time, average managers who desire to apply the MODSS to their daily operations encounter frustration due to their lack of in-depth knowledge of the MODM algorithms. In order to avoid misuse and even non-use of the computerized MODSS, this system must be capable of guiding the user possessing various levels of expertise in the use of these methods. Thus, a knowledge base can be defined as an intelligent guide (or front-end) to help the decision makers in selecting the suitable MODM method for each decision problem, and should be utilized to provide guidance on the selection of the right strategies for analysing their data. Once one method is selected, the system will capture the data needed to run this decision method, and then interact with users to obtain decision alternatives. The detail of this point will be discussed in Chapter 6.

4.3.2.5 Supporting the Solutions Analysis Procedure

The knowledge embedded can also be used either to direct the exploration process in the solution set or to support the finding of more efficient solutions in an MODM method (Pomerol 1993). Once an MODM method has been selected and an MODM application model has been defined, it is important to be able to use them easily for finding and analysing solutions. When studying a set of efficient solutions, it is very useful to be able to distinguish in the model the following: decision variables, goals, objectives and constraint functions. The usual intelligent assistance that an MODSS will provide to support solution finding is the computation of the variables and changing of the objectives of the model as the algorithm progresses.

4.3.2.6 Supporting Solution Evaluation

In most of MODM the output is not a choice, it is an ordering of efficient solutions. An MODSS should support an evaluation of efficient solutions according to the decision makers' desire for objectives, their goals, weights and priorities. In MODM problems, every decision maker is confronted with the necessity to make complex calculations. The widespread use of spreadsheets has shown how large this need is. In the domain of MODM, a certain number of classical and fuzzy criteria can be used for evaluating alternative solutions. To decide among several alternatives the decision maker usually considers relevant criteria. These criteria may be numerous and are usually conflicting. Most of the time in real decision making, the selection of one solution is often done intuitively.

Usually, the process of building the users' solution evaluation models comprises two tasks: determining satisfactory objectives to include in the model and finding a vector of decision variables. Pomerol (1993) showed the use of an ES to build a symbolic utility function in MCDM. A similar structure, which consists of an ES processing various data to access the value of different objectives, can be applied in MODSS. The main criticism of this technique is that it merges choice and descriptive items. To overcome this difficulty and to separate the choice (preference) from the descriptive (factual) part of the objectives, Levine et al. (1990) provided for the imagination of a structure in which each objective is characterized by a fact (base) and each constraint by a rule base. The rule base is organized as a semantic tree of constraints. The evaluation of an objective is made by applying the rules to describe the facts characterizing each objective.

4.3.3 A Functional Framework of IMODSS

An integrated view of IMODSS can be best illustrated by adding a new component part in the functional framework of MODSS, namely, an *intelligent guide component.* This component enables MODSS builders to move problem domain knowledge/information from the human (experts) to the computer to provide the support in one or more aspects mentioned above: man-machine cooperation, problem identification, information access, MODM methods selection, solution analysis and evaluation. These knowledge bases can respectively increase the effective use of MODM methods, enable MODSS not only answer what if questions, but also be able to answer why, and make MODSS more active and potentially more valuable. Usually, an MODSS only contains one or two of these knowledge bases. The functional framework of IMODSS is shown in Figure 4-8.

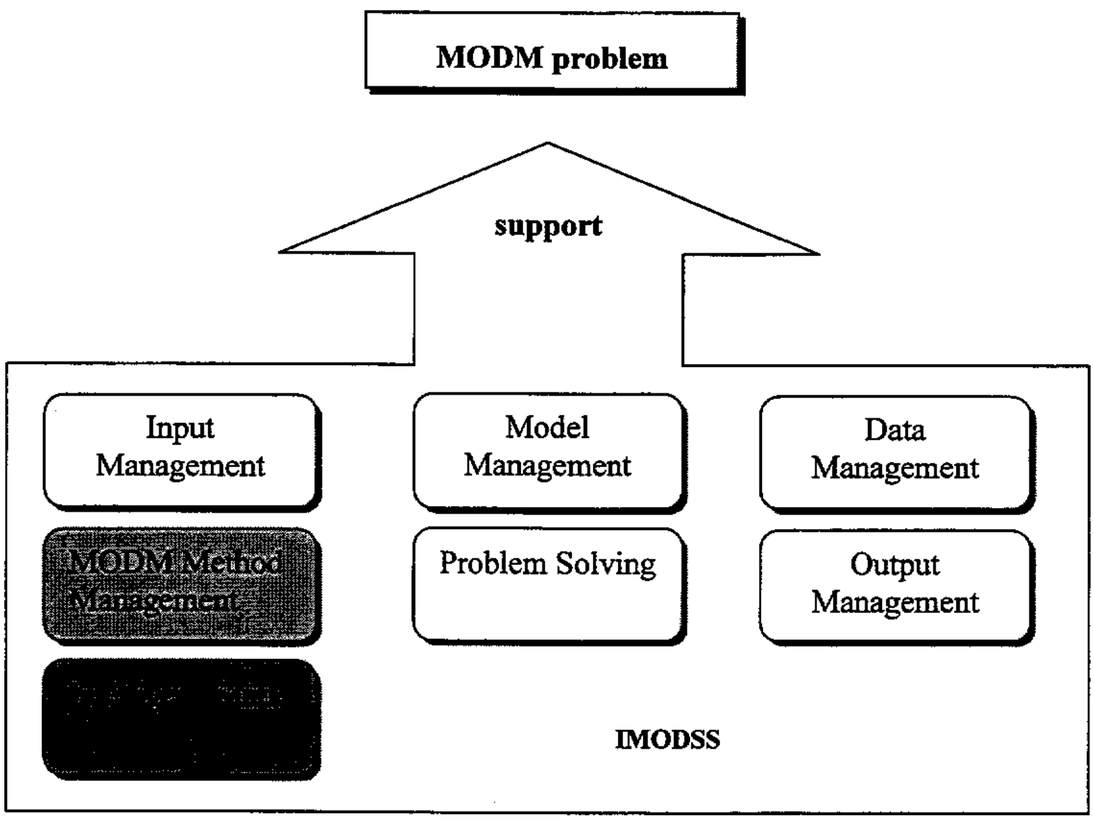


Figure 4-8 A functional framework of IMODSS

4.3.4 An Application Framework of IMODSS

When knowledge has been formalized under the form of rules, for example, to constitute a knowledge base, it is possible to request the system to reach a conclusion based on what can be logically derived from the knowledge stored under the form of rules and observed facts. Therefore, this intelligent guide component in Figure 4-8 will be implemented by a knowledge base that contains qualitative and symbolic knowledge represented using rules, frames or semantic nets, in addition to the factual data found in a database (Benbasat & Nault 1990). This knowledge base will need inference engines providing generalized reasoning methods, such as forward and backward chaining, and explanation facilities as to why a certain piece of information is requested, or why a given action is taken. This knowledge base, including an inference engine will be linked to the interface component in order to make up a system query user portion and an explanation portion. Figure 4-9 shows an application framework of IMODSS with a knowledge base and its physical components converted from the function framework shown in Figure 4-8.

4.3.5 Multiple Layered Conceptual Framework of IMODSS

A specific conceptual framework of IMODSS can be further depicted as a multiple layered hierarchy as reflected in Figure 4-10. A knowledge-based intelligent guide in this system is applied to support decision makers to "select a suitable MODM method" to match their application requirement and their preferences. There are six layers in this multiple layered conceptual framework (Lu, Quaddus & Williams 1999a).

The bottom of the hierarchy is the MODM methods (algorithms) for generating solutions to decision problems. A sufficient number of methods should be contained to increase the flexibility and usability of the system in order to allow the decision makers to use the most efficient methods for each particular decision problem or to resolve complex problems that could not otherwise be solved with a single MODM, or to allow the decision makers to get solutions from different methods.

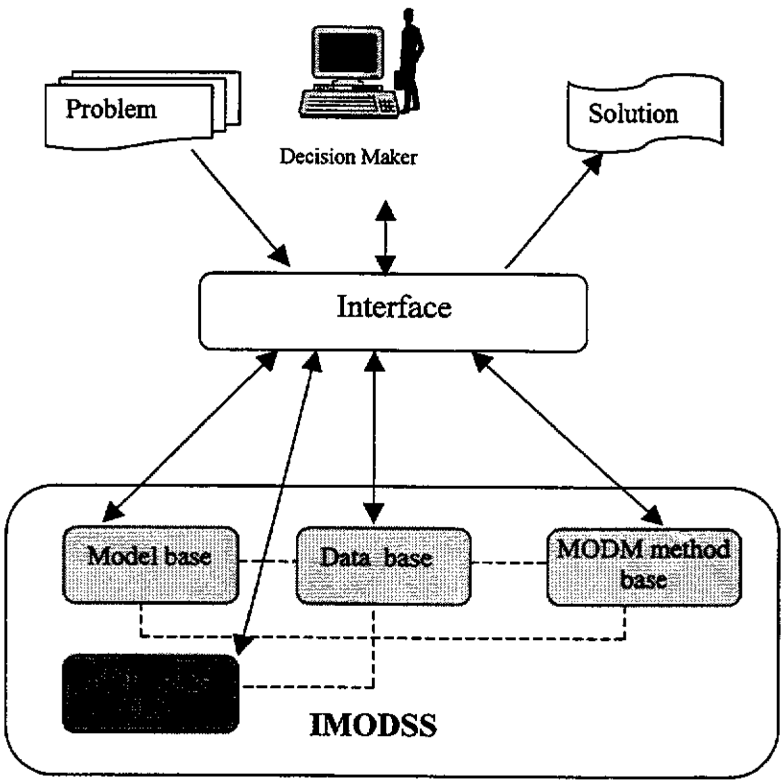


Figure 4-9 An application framework of IMODSS

The next layer is resource layer, which includes the database, model base and knowledge base. The resource layer must be able to be adapted with updated data files, models, facts, and rules, from time to time.

The next layer comprises the various distinct techniques decomposed in a generic fashion within the domain of an application framework. These include integrating an expert system with DSS and running various MODM methods in an interactive fashion.

The next layer is the function layer, representing fully operational subsystems, such as data operation, model operation knowledge, rule management, report management and other input/output management subsystems.

The next layer represents the guidance layer. In order to avoid misuse and even non-use of the computerized MODSS, this system must be capable of guiding the user possessing various levels of expertise in the selection and use of these methods. An intelligent component is executed to provide the guidance to the decision makers through an inference process.

The highest layer in the hierarchy represents the interface sector where the decision makers will use the object oriented graphical interface.

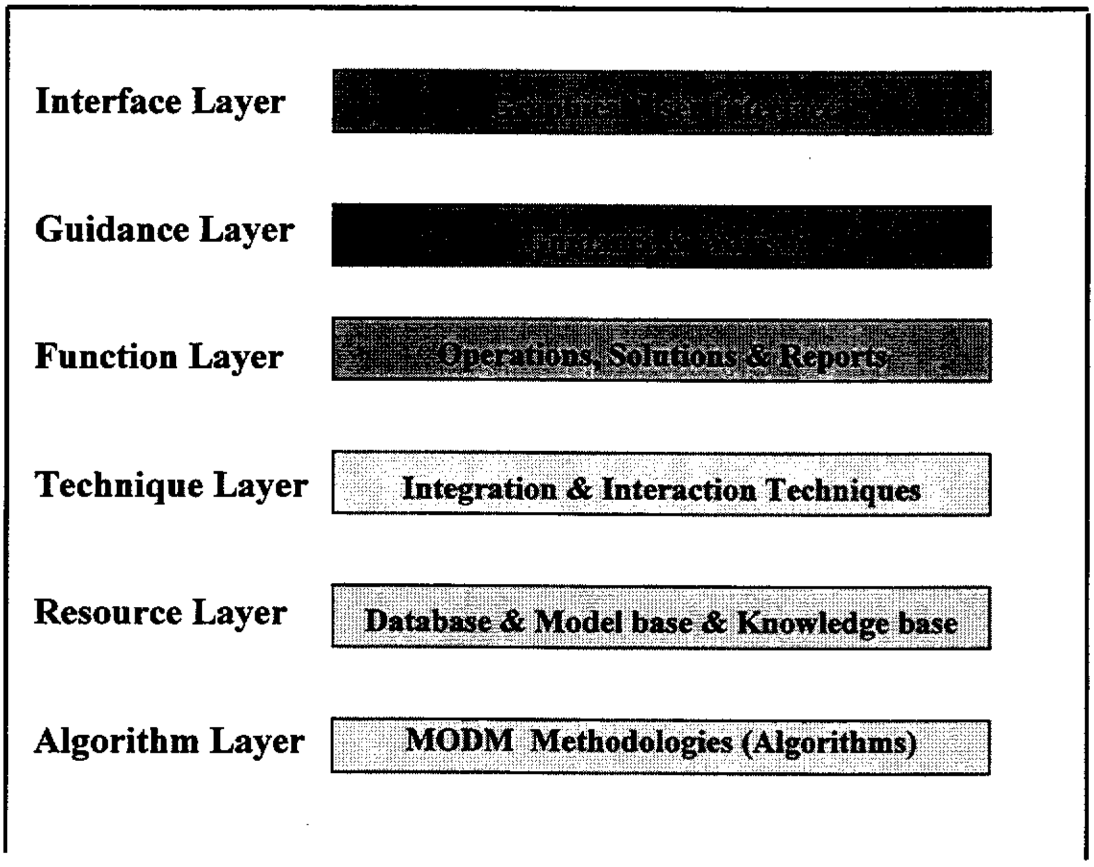


Figure 4-10 Multiple layered conceptual framework of IMODSS

4.4 Framework of Multiple Objectives Group Decision Support Systems (MOGDSS)

4.4.1 Framework of Group Decision Support Systems (GDSS)

4.4.1.1 GDSS Assumptions

A group decision making (GDM) process can be viewed as a coordinated decision making effort between communicating individuals who each possess specialized knowledge in order to solve a complex business problem. A decision making group is defined as two or more people jointly responsible for a decision task, not necessarily situated in the same physical location, and who perceive themselves as part of a team working on the task (DeSanctis & Gallupe 1987). A GDM problem can be characterised by an MADM model, that is, a small number of distinct alternatives evaluated on a number of group attributes. Mathematically, the GDM problems can be represented as follows:

$$\left.\begin{array}{l} \textit{Select: alternative } A_1, \textit{alternative } A_2, \ldots\ldots \textit{alternative } A_m \\ \textit{Subject to: attribute } C_1, \textit{attribute } C_2, \ldots\ldots \textit{attribute } C_n \end{array}\right\} \qquad (4.2)$$

where $A = (A_1, \ldots\ldots, A_m)$, denotes m distinct alternatives, $C = (C_1, \ldots\ldots, C_n)$, represents n group decision aggregation attributes.

As discussed in section 2.6, group decision support system (GDSS) is an extension of the DSS concept to decision making groups (Dennis et al. 1988), and is also a sub-field within DSS (Sprague, 1995). Definition of GDSS has been offered by DeSanctis in 1985. DeSanctis defined GDSS as "an interactive computer-based system which facilitates the solution of unstructured problems by a set of decision makers working together as a group". Sage (1991) further explained GDSS as an information technology-based support system designed to provide the means to conduct group interactions in a particular decision making group and a particular GDM process. The deployment of GDSS improves the GDM process by providing tools for use in communication, computing and decision analysis group techniques. The development of GDSS also provides new approaches for organization-wide decision making and facilitates unstructured problem solving (Radermacher 1994).

Some broader definitions of GDSS, however, are not high level limited to decision making (Palmer, Fields & Brouse 1994). However, this thesis confines our discussion to the decision making aspects of GDSS and is based on the following assumptions:

- There are multiple decision makers to form a decision making group.
- The geographical locations of the decision makers can be dispersed.
- The decision makers interact in a co-operative manner and in a trusting environment.
- The group shares a common set of feasible decision alternatives.
- Each decision maker has his/her personal objectives or goals that reflect a priori values and aspiration levels.
- When a consensus is not found, negotiable alternatives are co-operatively sought.

4.4.1.2 Classes of GDSS

DeSanctis and Gallupe (1987) outlined three classes of GDSS. Each class improves on the previous classes in terms of providing the means for group communication and therefore requiring the support of more sophisticated information technology.

- The first level includes GDSS that provides features to remove physical communication barriers (i.e. electronic mailboxes, broadcast messages) in the group in order to facilitate information exchange. The resulting GDSS plays a passive role in the GDM process. That is, the degree of change in communication channels is minimal.
- The second level of GDSS provides decision modelling, group decision techniques and qualitative modelling tools available to the entire group, such as multiple

objective decision analysis, thus reducing the degree of uncertainty in the decision making process. The deployment of such a system would also result in a greater degree of impact in the way that inter-group communications are conducted.

- The third level of group support provides machine-induced communication patterns and controls the communication process by imposing communication patterns within the group. In this level, the data integration is implemented and judgements are synthesized. Rules and constraints are employed to control the development of such patterns, timing and most importantly the content of this information exchange in a group interaction procedure (Rao, Sridhar & Narain 1994).

Benbasat and Konsynski (1988) have pointed out that advancement in the development of a third level GDSS is a necessary ingredient in achieving true group decision support. The development towards a third level GDSS not only signifies the more active role the GDSS plays in a GDM process, but also imposes the requirement that any future GDSS must have a greater understanding of the group activities and any goals that might arise during such activities (Werners 1987). With an increased understanding of group interactions by way of presenting a GDSS framework, further GDSS development efforts can be to design and implement a GDSS that plays a more active role in controlling group activities. The ultimate goal of true group decision support can only be achieved with a framework that provides the theoretical basis for future GDSS to fully understand the various group interactions (Wang 1997).

4.4.1.3 A Functional Framework of GDSS

A major objective of a GDSS is to support the group decision making process through the organization of decision criteria and alternatives according to a group decision model. Since the emphasis of a typical GDSS is on the support, it must focus on the understanding and improving the group decision making process for those decisions that are commonly based on the interaction among a group of members.

GDSS can be viewed as subsuming conventional DSS within it. That is, the concepts of model, database, and human interface all apply. Thus, as group size to one, a GDSS reduces to a DSS (Sprague 1995). Conversely, in moving from a DSS to a GDSS, some new requirements are introduced: (1) the addition of communications capabilities; (2) enhancement of the method base to provide GDM, and so on for developing consensus; (3) greater system reliability; and enhanced physical facilities. Therefore, a GDSS is made up of the six components shown in Figure 4-11. Compared to the functional framework of MODSS, a major functional component in this figure is *group aggregation.* In a group decision making process, the preferences of the group members vary from each other, and they may apply different solutions and methods, and choose different solutions. Consequently, determining the best alternative solution to a decision problem requires an aggregation of individual preferences. This is especially true for an interactive procedure that requires group feedback to evaluate alternative solutions.

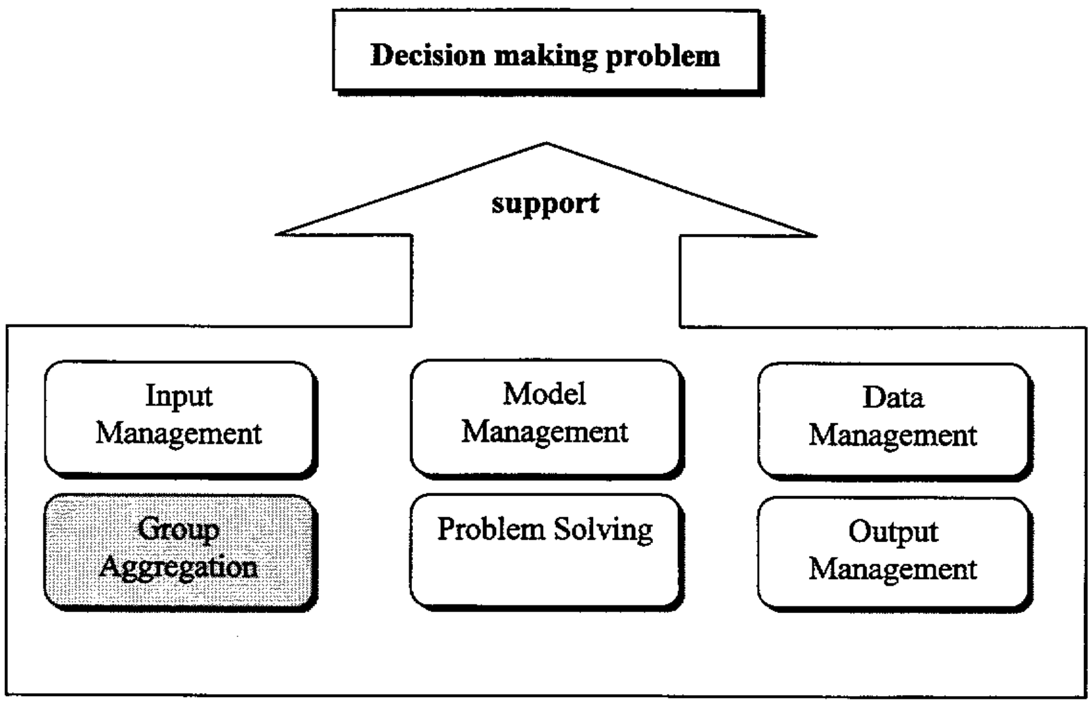


Figure 4-11 A functional framework of GDSS

4.4.1.4 An Application Framework of GDSS

In practice, a well-structured GDSS constitutes an information exchange network through different channels to assist managers in decision processes to improve the effectiveness of GDM. There are some parallels between these GDSS components and those of an MODSS. The database component in an MODSS is used to store data of multiple objective decision applications. The model base component consists of a group of users' application models. The MODM method base, generally, made up of several MODM methods, solves various MODM problems. The interface component controls the interaction between decision makers and the system. A GDSS is also made of up of four components: database component for group decision data, model base component for group decision application model, GDM method component for providing group aggregation method and interface, also as a component to support entire group decision interaction activities. Figure 4-12 presents a simplified GDSS application framework exhibiting the physical components and *n* decision makers in a GDSS.

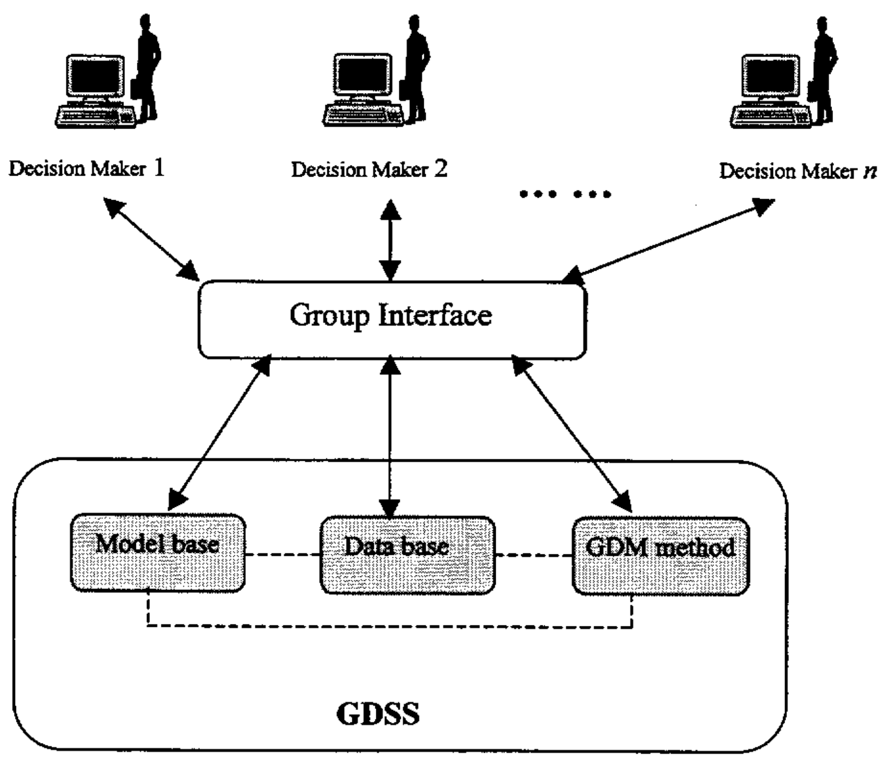


Figure 4-12 An application framework of GDSS

4.4.2 Functional Framework of MOGDSS

4.4.2.1 Two Working Patterns

An MODSS is considered to support the decision making process where a complex set of objectives must be analysed with respect to constraints in order to determine a satisfactory solution. This type of decision making is often taken in a group environment. Therefore, MOGDSS is defined as a GDM theory, and provided with MCDM methods, and DSS, and taken together, offer great promise in supporting multiple decision makers to reach agreement on decision involving multiple objectives under a framework of DSS (Iz & Gardiner 1993).

There are two approaches to reach an MOGDSS through the extension of an existing system. One is to integrate an MODM function into GDSS (MO+GDSS), that is, to enhance the method base to provide GDM. Another is to move MODSS to a group decision environment (MODSS+G), that is, to add a group aggregation function in MODSS. Previous researches have been found in the literature taking the first approach, such as Iz (1990). This research takes the second approach.

Two MOGDSS patterns are produced. The first pattern of MOGDSS integrates the MODM model and GDM model together. In this pattern of MOGDSS, the first step of decision support typically consists of generating a set of efficient solutions of MODM by applying an MODM method as alternatives for consideration by a group. The second step is to determine a best alternative solution from the set of efficient solutions through an aggregation process of individual preference (Iz & Gardiner 1993). This pattern is shown in Figure 4-13.

The second pattern of MOGDSS applies two separated methods: MODM methods and GDM methods. The system contains an MODM methodology base to support decision making.

In the first stage, each member can get an MODM solution through any MODM method from this methodology base under an MODSS form. In organizations, however, there are many situations where the goals of each group member may not be the same as the others, and not everyone may have the same information. Therefore, these decision makers' solutions are usually different from each other.

In the second step, these solutions consist of a set of candidate alternatives for the group. The group members then negotiate through the set of solutions so as to achieve a compromise decision solution by using GDM method under a GDSS (Kersten 1985). We summarize the main steps of this pattern in Figure 4-14. This research will follow the second pattern.

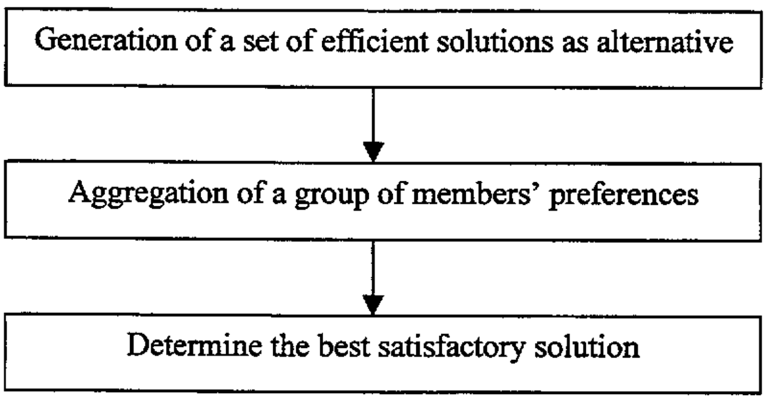


Figure 4-13 Pattern 1 of MOGDSS

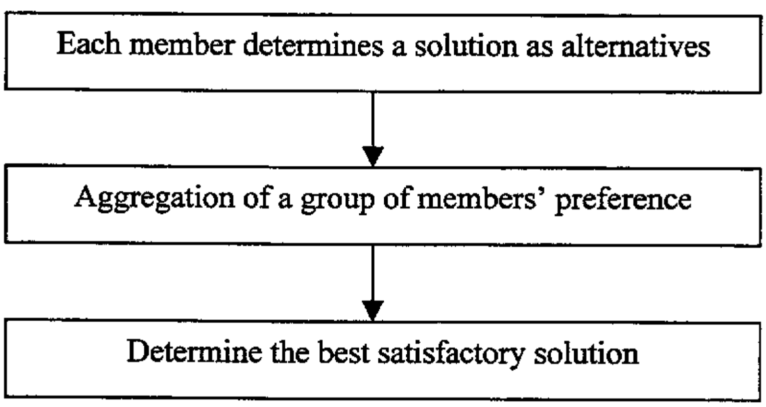


Figure 4-14 Pattern 2 of MOGDSS

4.4.2.2 Group Decision Communication Channels

An MOGDSS must facilitate the exchange of information and expertise between group members. Therefore, communication channel is an important aspect of MOGDSS. An analysis of face-to-face (FTF) versus computer-mediated-communication (CMC) channels was presented by Iz (1992). For the FTF communication channel, a decision room and a decision meeting are necessary, but for CMC, the need for a meeting or direct communication is unnecessary.

Although almost all group decision meeting today are FTF, technology is starting to be applied to make it possible for participants to be separated in space and/or time. These electronic meeting systems include computer conferences and video teleconferences. Table 4-2 shows the four possible combinations of proximity and separation in space and in time. This thesis will concentrate on one form of electronic meeting system, the 'decision room' in which the participants are in the same room at same time.

Table 4-2 Forms of electronic meeting system based separation in space and time (Sprague 1995)

	Same time	**Different time**
Same place	Example: decision room Need: face-to-face meeting Private displays Public displays	Example: Team room Need: Administration, filing, filtering Private workstations Group displays
Different place	Example: video conference Need: cross-distance meeting Conference calls Screen sharing	Example: computer conference Need: ongoing coordination Group conversations Group writing

Based on Table 4-2, Iz's analysis and Figure 4-14, there are four situations of communications available during multiple objective problem solving in a group environment. The four types of communication channels in MOGDSS (based on pattern 2) are shown in Table 4-3.

Table 4-3 Group communication channels of MOGDSS

Type	Communication channel (Step 1)	Communication channel (Step 2)
1	FTF	FTF
2	FTF	CMC
3	CMC	FTF
4	CMC	CMC

For an MODM application, type 1 requires that each group member uses an MODM method to get a solution in a decision room, that is in same time and same place. These members then are needed to select and use a GDM method to achieve a compromise solution through negotiation in a meeting. The situation of the type 3 is different, where each member can use any MODM method to get a solution in any time and any place. These members then come down to a decision room to select and use a suitable GDM method to achieve a compromise solution. For type 2 and type 4, the group aggregation can be taken at different times and in different places. In this situation, a coordinator will be needed. The coordinator receives every member' preference and assists this group decision process, but has no influence on the final solution of a group. The dynamic nature of the MOGDSS is achieved by a set of GDM methods and a network where each group member's computer is either directly or indirectly linked to each other to support the groups' activities under various channels. The interaction between the GDM methodology base and MODSS provides a framework for the group members to access relevant problem-solving information continuously rather than only at meetings. A more detailed discussion of these will be given in Chapter 8.

4.4.2.3 Functional Framework

An MOGDSS has been designed as an interactive computer-based system that use MODM models and methods to provide a means of analysing data, aiding in the choice between efficient solutions, and evaluating the performance of the satisfactory solution chosen by a group of decision makers. In an MOGDSS environment, each problem has multiple conflicting objectives. Each decision maker must generate relevant goals, weights for each objective of this problem. Each member of a group can use one or several different MODM approaches and techniques to arrive at a solution of this decision problem. Each of the other decision makers may share some, none or all of one decision maker's goals, and weights for these objectives. The problem is no longer the design of the most preferred solution according to one individual's preference structure. The analysis must be extended to account for the conflicts and aggregation among different group members who have different preferences for objectives. Therefore, the group of decision makers should agree with certain rules. The GDM can be usually understood to be the reduction of different individual preferences among objectives in a given set to a group collective preference. Therefore, compared to the framework of MODSS, a new component *group aggregation* is included into the functional framework of MOGDSS shown in Figure 4-15.

4.4.3 An Application Framework of MOGDSS

GDSS provide software to support the individual and to support the group. To allow each individual to do private work, the usual collection of text and file creation, graphics, database, and help routines are provided at the individual workstations. For the group as a whole, a GDSS provides software for summarizing the group's opinion. For example, the public screen can be used to present a cumulative list of all suggestions (a list of solutions for an MODM problem, in MOGDSS) and to show the aggregated results. As discussed later in Chapter 8, group members' solutions and preferences can be either identified by individual or aggregated so that individuals need not expose themselves if they view contrary to those of the senior person present.

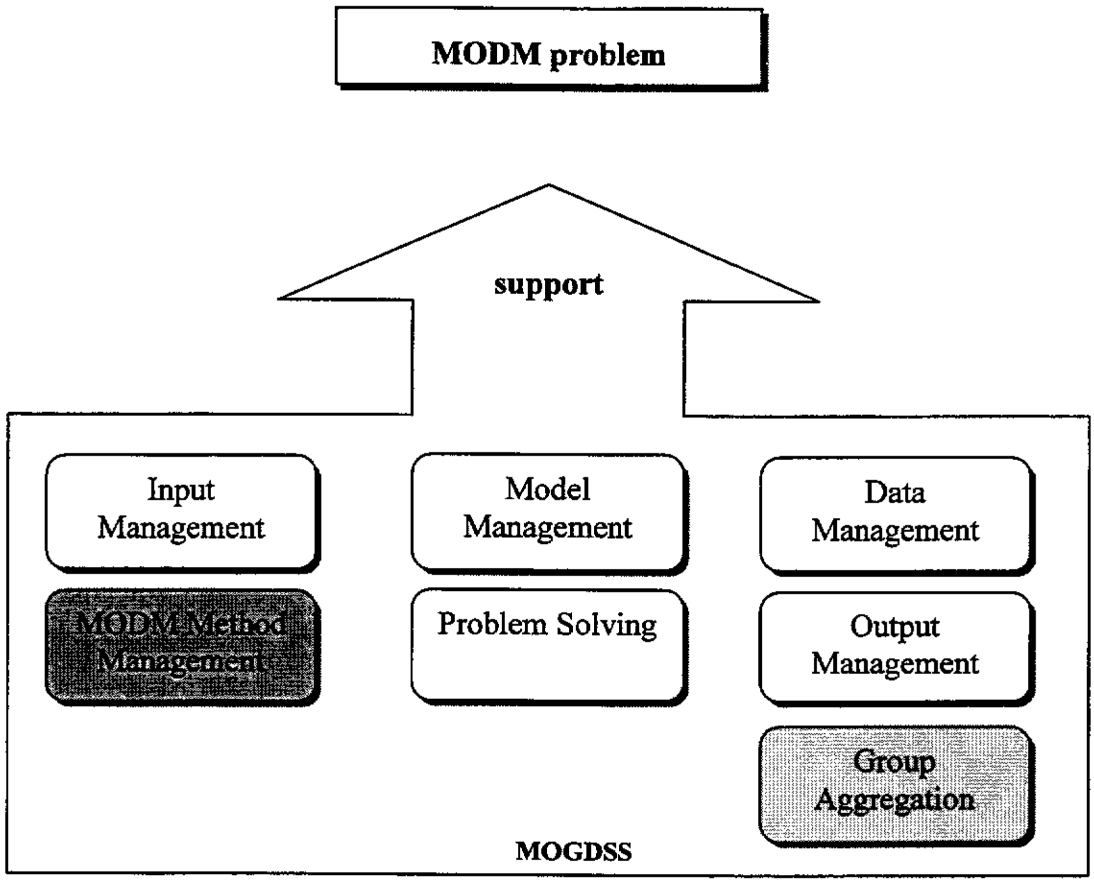


Figure 4-15 A functional framework of MOGDSS

The application framework shown in Figure 4-16 has two procedures: an MODM procedure and a group procedure. The MODM procedure involves the solution of an MODM problem by each individual decision maker. The relevant database, model base and method base are also considered in this procedure (procedure 1). This procedure represents every member's judgment for a satisfactory solution based on a set of conflict functions. The group members' different solutions are taken as alternatives for the group decision. The second procedure (procedure 2 of Figure 4-16) is used in order to reach a consensus. A GDM methodology base for aggregating group members' preferences is applied to finalise the decision solution. Various GDM methods are used to support different styles (i.e. interactive) and forms (i.e. different times and same place) of group meetings.

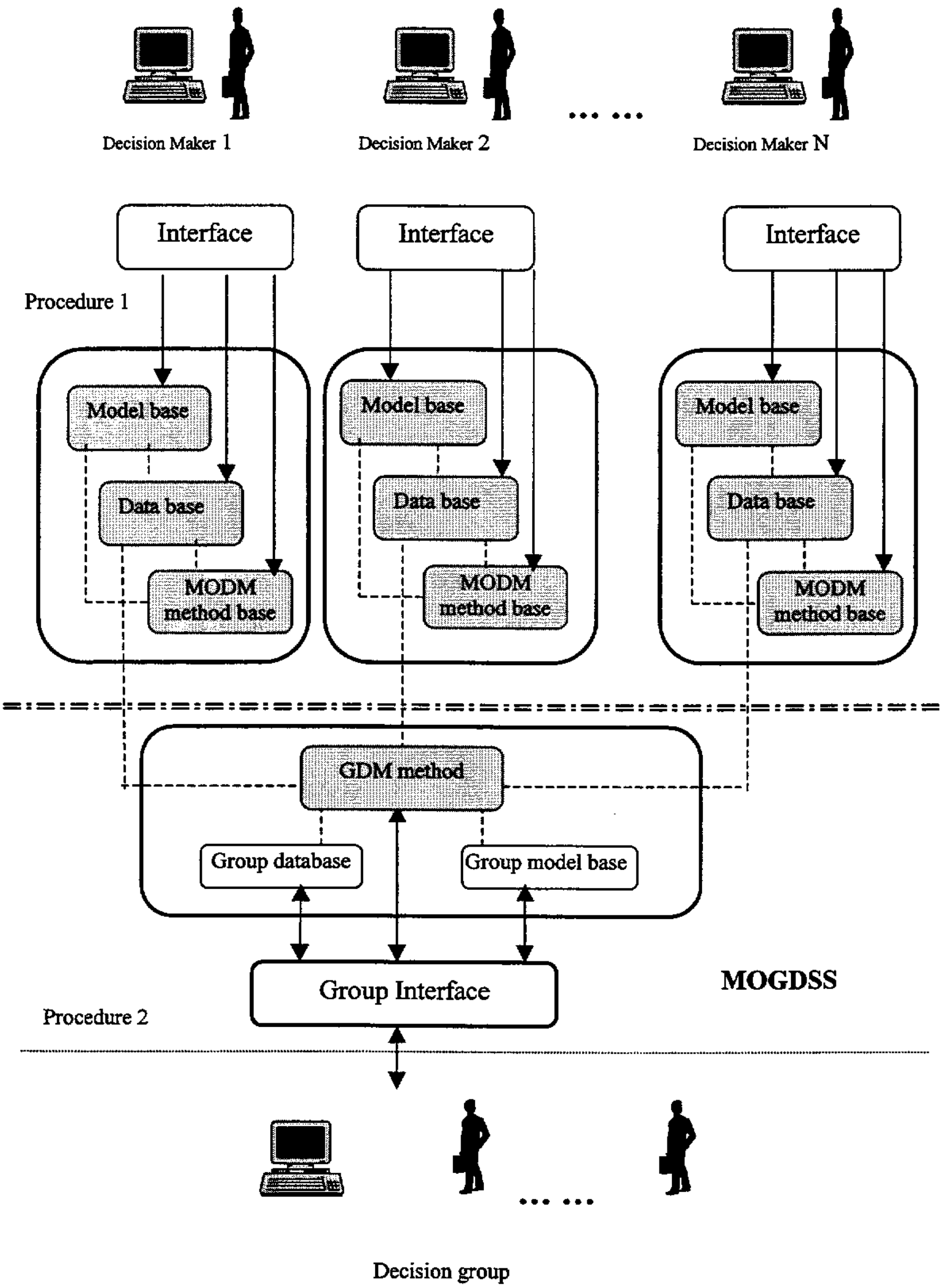


Figure 4-16 An application framework of MOGDSS

4.5 Framework of Intelligent Multiple Objectives Group Decision Making Systems (IMOGDSS)

4.5.1 Functional Framework of IMOGDSS

IMOGDSS is characterized as an interactive computer-based information system which combines the capabilities of MODM, knowledge-based intelligent technologies, group aggregation and communication technologies, database technologies, and decision technologies to support the identification, analysis, formulation, evaluation and solution of multiple objective problems by a group in a user-friendly computing environment.

IMOGDSS is expected to have higher mean quality and effectiveness values for all characteristics of MODSS, IDSS and GDSS. Therefore, it contains all functional components of the three systems. Comparing with the framework of MOGDSS, a new component *intelligent guide* is added in this framework as shown in Figure 4-17.

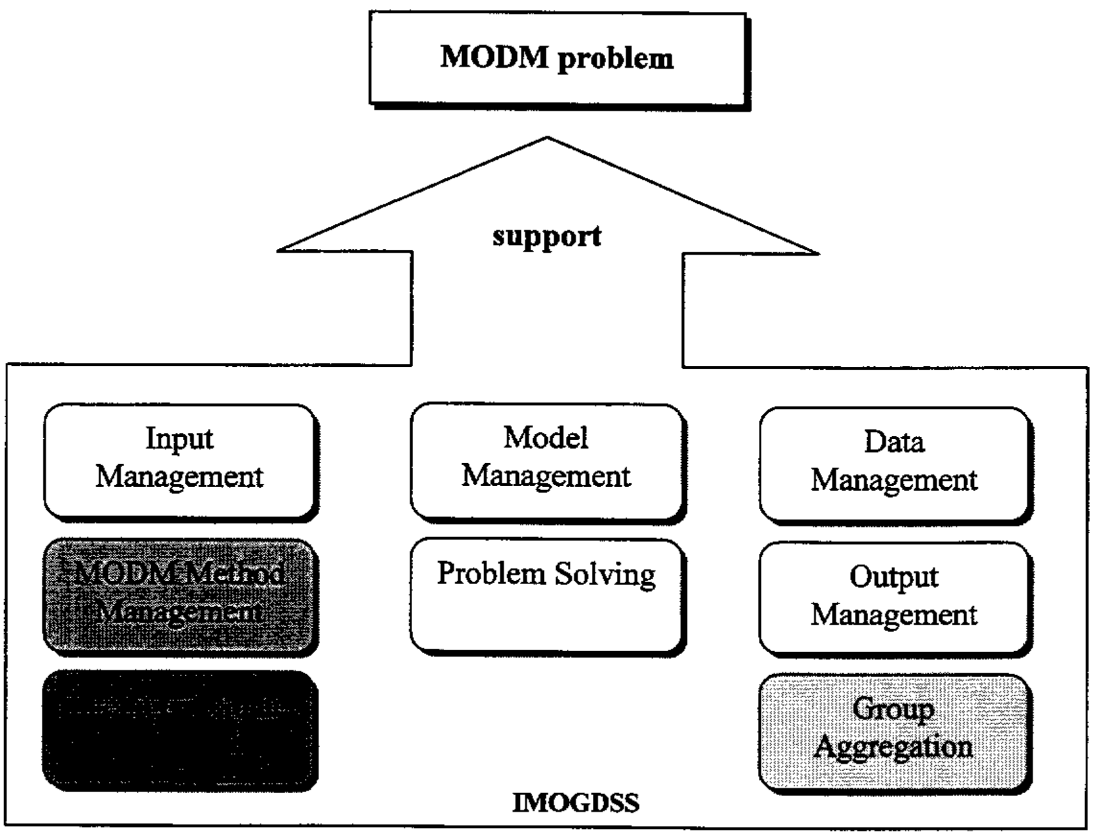


Figure 4-17 A functional framework of IMOGDSS

4.5.2 Application Framework of IMOGDSS

As with MOGDSS, the decision process in IMOGDSS for an MODM problem is also divided into two stages. In the first stage, the individual stage, each decision maker makes a decision through a series of interactive operations by using any MODM method in MODM methodology base. They will share data and models involved in this application. The group of decision makers' solutions generate a set of alternative solutions for considerations by the group. However, the difference from MOGDSS is that a knowledge base is integrated in each individual decision system to increase problem solving effectiveness. The second stage is allocated in a group environment where decision makers' solutions are aggregated through a group aggregation method in the GDM methodology base so as to achieve a best compromise decision. A knowledge base can also be added in this group aggregating process to support the group decision methods' use. Figure 4-18 shows the two stages. It must be noted that:

- there is an MODM methodology base to support decision makers' decision making; each member can get an MODM solution through same or different methods from the methodology base;
- not all members of the group have the same knowledge of MODM and thus they can use the intelligent guidance system with different levels;
- members of the group have conflicting objectives for a decision problem, they can work in a decision room or different locations; and
- there is a GDM method base, group database and group interface to receive and store all solutions from each member, to analyse and evaluate these solutions, interact with group members and display the final solution of this group.

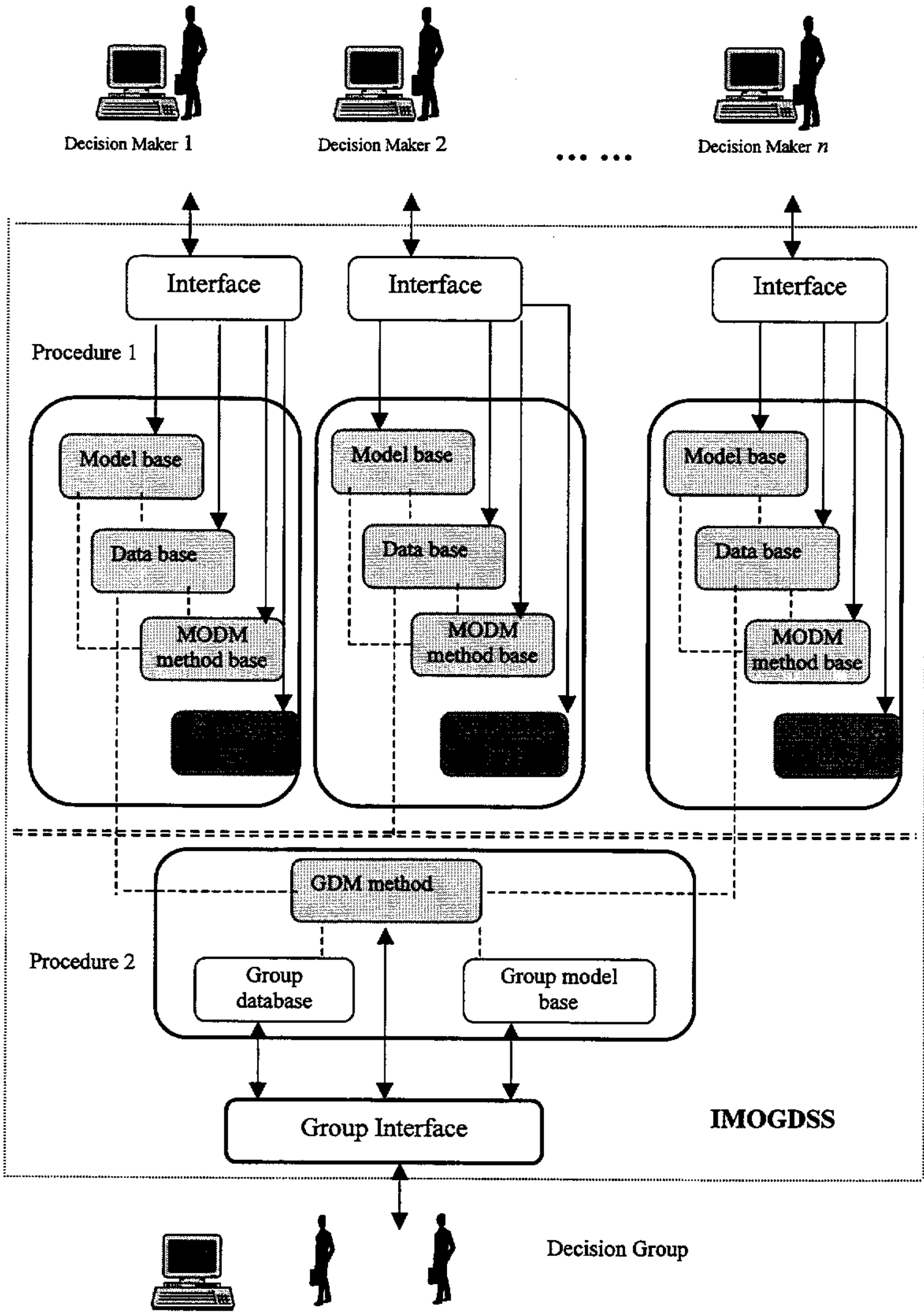


Figure 4-18 An application framework of IMOGDSS

4.5.3 Factors for IMOGDSS Integration

Integration has been a main strategy for implementing IMOGDSS. Integrating these computer-based information systems with one another will enhance the quality and efficiency of many computerized systems in total (Turban 1995). The integration of IMOGDSS is taken in both functional integration and resource integration. Therefore, the integration in IMOGDSS means that the systems are integrated into united facility rather than having separate software, data sources and communications for each independent system, and it implies that different support functions are provided as a single system. Many factors should be considered when IMOGDSS is integrated.

Architecture of integration. Several alternatives are available for executing the integration. Each of these options has some benefits as well as costs and limitations. A careful analysis should be undertaken before the integration.

Development process. The development process of many information systems projects follows a sequential life-cycle approach. In contrast, most DSS and IDSS are prototyped.

Data structure issues. AI applications are centred around symbolic processing, whereas DSS, GDSS and MODSS projects are built around numeric processing. When these systems are integrated, data will have to flow from one environment to a different one (Turban 1995). Databases are structured quite differently from knowledge bases. In a knowledge base, procedural information and declarative information are separated, whereas in a database everything is combined. It is easy to develop a conceptual system with a database and knowledge base and show that the two are interconnected. To implement the interconnection is a challenging task.

Connectivity. AI applications may be programmed or special knowledge engineering tools may be connected, or a combination of them. The shells may be written in different computer languages, but not necessarily in the same language that was used to write the DSS that is being integrated with the AI part.

4.5.4 A Formal Mathematical Description of IMOGDSS

It has been discussed that an MODM model can be built by considering a vector of decision variables, developing a set of objective functions and constraints. It has been also described that a GDM problem can be characterised by a small number of distinct alternatives evaluated on a number of group attributes. Each decision maker always attempts to maximize the objective functions under the constraint functions and their preferences. And a group decision' selection is normally based on maximizing a multiple group criteria value (or utility) function elicited from the stakeholders. Therefore, a multiple objectives group decision making (MOGDM) can be described as p decision makers to determine an optimal decision that optimizes n objectives subjected to m constraints under q group criteria. Mathematically, the MOGDM problem can be formulated as follows:

$$\left.\begin{array}{lll} Select: & alternative & S_i,(i=1,2,...p) \\ Subject\ to: & criteria & C_j,(j=1,2,...,q) \end{array}\right\}$$

S_i ($i=1..p$) is Di' solution for the MODM model: (4.3)

$$\left.\begin{array}{ll} Maximize & \{ f_i(\overline{x}),\ i=1,2\ ......,\ n \} \\ Subject & to\ : g_j(\overline{x}) \le b_j,\ j=1,2\ ....,\ m \end{array}\right\}$$

where

$S=(S_1,\cdots,S_p)$, denotes p solutions chosen by decision makers as alternatives (supposing $S_i \neq S_j$ if $i \neq j$) ;

$C=(C_1,\cdots,C_q)$, represents q group criteria;

$\overline{x}=(x_1,x_2,\cdots,x_k)$ denotes k-dimensional vector of decision variables;

$F=(f_1(\overline{x}),\cdots,f_n(\overline{x}))$ represents n objective functions;

$G = (g_1(\overline{x}), \cdots, g_m(\overline{x}))$ represents *m* constraint functions;

$D_1, \cdots, D_p$ are decision makers.

An IMOGDSS can be designed as

$$IMOGDSS = \{KB, MB, GB, DB, OM, GUI, CC, D\}$$

to support the model (4.3). Where

KB—a knowledge base

MB---an MODM methodology base

GB---a GDM methodology base

DB---database

OB---model base

GUI---graphical user interface

CC---communication channel

D---the set of decision makers

Therefore, an IMOGDSS working process is described as:

For any $D_i \in D$ $(i=1, 2 \ldots p)$ selecting a suitable method M_l $(l=1, 2 \ldots w) \in MB$ guided by *KB*, accessing to *DB*, and forming a model $MO_k \in OB$. A solution $S_j \in S$ $(i=1,2 \ldots p)$ is obtained through an interactive process supported by GUI. A $G_n \in GB$ is selected to evaluate S_i $(i=1 \ldots p)$, and help D_i $(i=1, 2 \ldots p)$ arrive a final compromised solution S_{j_0}.

4.6 Summary

This chapter attempts to investigate and build a functional framework and an application framework of IMOGDSS where MODSS, IDSS and GDSS are integrated to reduce communication barriers, enhance group capabilities, improve the effectiveness of MODM problem solving, increase the depth of decision analysis, and lead to greater satisfaction in the multiple objectives group decision making process.

These frameworks produced in this Chapter, collectively termed as interpretive structural framework, can aid in the development of IMOGDSS through:

- creation of a structure which helps to identify the presence or absence of information necessary for decision making,
- definition of levels in the system hierarchy,
- specification of subsystems at each level, and
- facilitation of user participation in the design process.

Chapter 5

PROTOTYPE OF INTELLIGENT MULTIPLE OBJECTIVES GROUP DECISION SUPPORT SYSTEMS [3]

5.1 Introduction

This chapter presents a GUI based design and implementation of an integrated intelligent multiple objectives group decision support systems (IMOGDSS) prototype which allows selective and flexible use of many popular MODM methods to solve a multiple objective decision problem under a knowledge-based intelligent guide. The system then aggregates a group of decision makers' solutions using one or more group aggregation method(s) in GDM methodology base to arrive at a compromised solution. This chapter illustrates ways in which the conceptual framework described in Chapter 4 can be used in practice. As suggested in the previous chapters, other MOGDSS also are able to apply the framework in the way that is most appropriate to their own situation.

[3] This Chapter is based on papers presented at several conferences:

(1) Lu, J. M.A. Quaddus and R.Williams, " Developing a knowledge-based multi-objective decision making system", *Proceedings of Hawaii International Conference on System Sciences (HICSS),* 4-7, January, 2000.

(2) Lu, J. M.A. Quaddus and R.Williams, " The design of a knowledge-based guidance system in IMODSS", *Proceedings of Australasian Ccoference on Information Systems,* Wellington, New Zealand, 1-3, December,1999.

This chapter is organized as follows. Section 5.2 introduces the architecture of the IMOGDSS prototype. Section 5.3 proposes the design of interface, including input subsystem and report subsystem. The design of MOGM methodology subsystem is provided in Section 5.4. Section 5.5 describes the structure and working principle of the knowledge-based intelligent guide subsystem. Section 5.6 highlights the design of the database and model base subsystem. A group subsystem with a GDM methodology base is proposed in Section 5.7. In the last section a summary is given.

5.2 Architecture of IMOGDSS Prototype

An integrated software package called IMOGDSS is designed and developed as a prototype in order to implement the framework provided in Chapter 4. It is essentially applied for modelling and solving multiple objective linear programming (MOLP) procedures under a single decision maker or a decision group environment. To utilise the potential of the MODM methodology base effectively, this IMOGDSS is designed to include seven popular MODM methods and has the capability of guiding the decision makers to select and use the most suitable MODM methods from the seven methods for solving their decision problems. A knowledge-based intelligent guide subsystem is provided to achieve the aim. Therefore, this system shows how a decision maker's preferences are identified, how an MODM method is recommended and how a multiple objectives decision problem is analysed and solved. This system also can be used in a decision group. After each decision maker of a decision group makes a decision for an MODM problem, a group subsystem is launched to exchange information and ideas for objectives and their goals, and to identify acceptable and desirable solutions. Usually a negotiation about their solutions is processed so that this decision group achieves a compromise solution of this MODM problem. A GDM methodology base that consists of several group aggregation methods is utilised to find a compromise solution.

Integration is one of the principal problems facing designers of DSS development tools (Klein & Methlie 1995). Taking both resource integration and function integration, IMOGDSS integrates a database for original data and solutions, a methodology base of seven MODM methods, a model base for users' models, a knowledge base that is utilised to provide guidance on the selection process of suitable MODM methods according to different decision problems and the decision makers' situations, and a GDM methodology base for supporting a group aggregation procedure of the solution of MODM in a decision group. The overall architecture for the IMOGDSS is shown in Figure 5-1. Eight major subsystems namely, interface subsystem, input subsystem, intelligent guide subsystem, method subsystem, data subsystem, model subsystem, report subsystem and group subsystem are contained in this IMOGDSS.

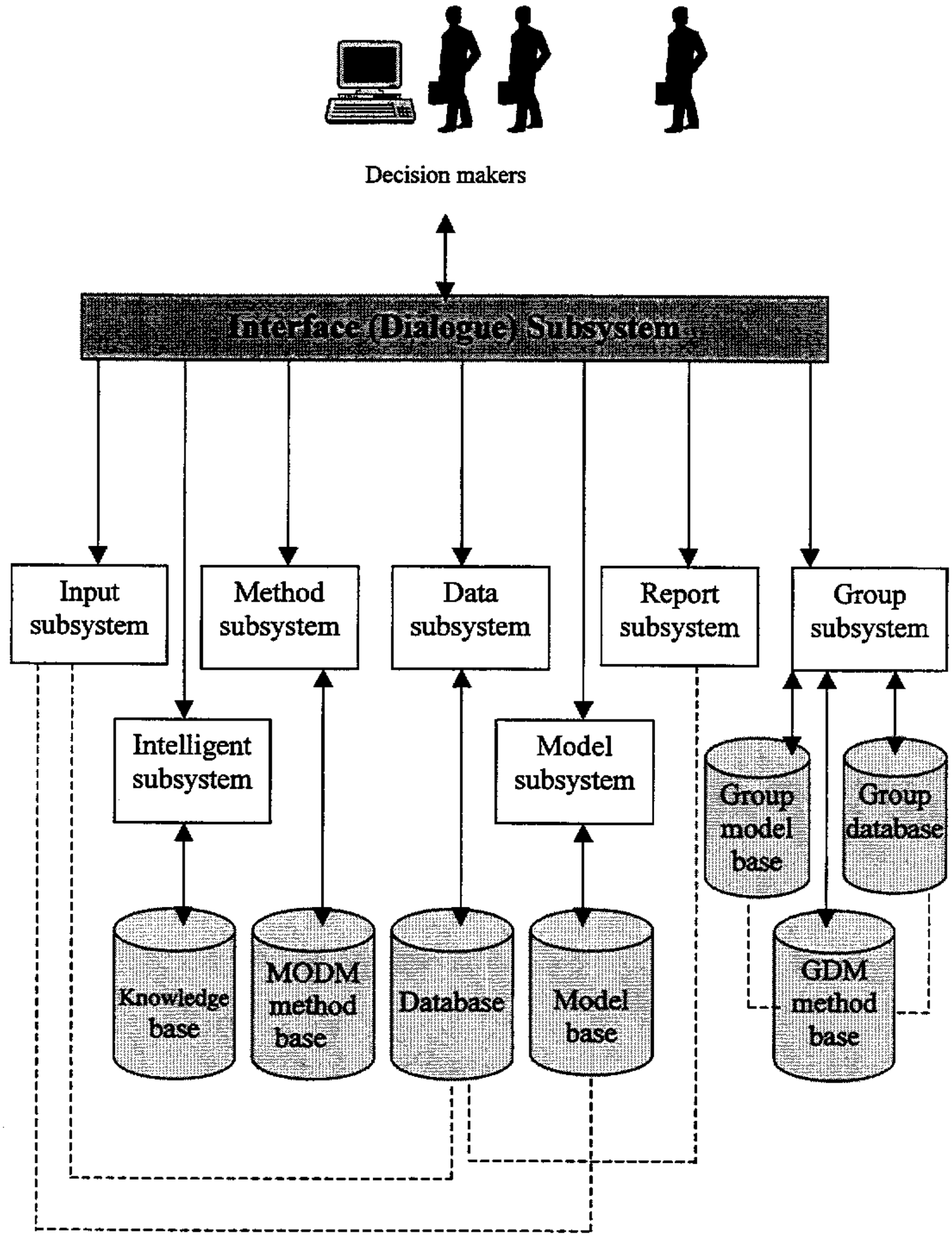


Figure 5-1 Architecture of IMOGDSS prototype

In this IMOGDSS, each subsystem is linked with the interface subsystem to incorporate a united communication interface between the decision makers and the system, and different resources are interconnected. Each subsystem is also connected with one or more resource base(s) so that various information sources can be accessed. The MODM methodology base, knowledge base and GDM methodology base are three main components of the IMOGDSS to support the multiple objective problem solving. This IMOGDSS has been implemented as a menu-driven software package in which the system presents its queries to the user in the form of a menu with alternate selections. DELPHI is used to implement this prototype. An expert systems shell called CLIPS is also embedded in this system for the knowledge based programs and compiles them into runtime executable files. The project source of this IMOGDSS is shown in Appendix 1.

This IMOGDSS has a good applicability and broad suitability for applications in various decision situations. It can be applied to solve MODM problems which have a maximum 35 decision variables, 20 constraints and 10 objective functions. It can be used by an individual decision maker or a decision group with up to 30 decision makers.

5.3 Design of GUI-based Interface Subsystem

5.3.1 Main Interface

The IMOGDSS prototype mixes different kinds of interfaces: windows, menus, dialog boxes, icons and forms. It is essential to be able to apply a GUI and graphical display function which will provide an assistance function of MODM models, GDM models and their solutions to aid decision makers to understand, analyse and choose from them. The interface subsystem serves to integrate various other subsystems as well as to interact with the decision makers. The main interface consists of a system desktop with a pull-down menu bar at the top (Figure 5-2). There are nine sub-menus that form the functions of the user interface. They are File sub-menu, Input sub-menu, Intelligence guide sub-menu, Method sub-menu, Model sub-menu, Result Database sub-menu, Report sub-menu, Group sum-menu and Help submenu. These pull-down menus together with their respective windows perform all kinds of decision support activities.

The user interface component is the link between decision makers and the system. This subsystem is very important, because it is one that is seen and used directly by users, so the users may think this is the system. It should be a self-adaptive interface that automatically adjusts to users' preference and tasks, and provides the functionality and the form required to match the interface to a specific user performing a specific task. For example, the system processes the user's response and may either generate another menu to present further queries to the user or display a result. If a response points out that none of the listed options is acceptable, the user is presented with an edit window. The user can give a new value by simply typing in the response. This interface should also promote ease-of-use and consistency of features in the interface that are important factors in establishing usefulness and success of the system (Fazlollahi, Parikh & Verma 1997).

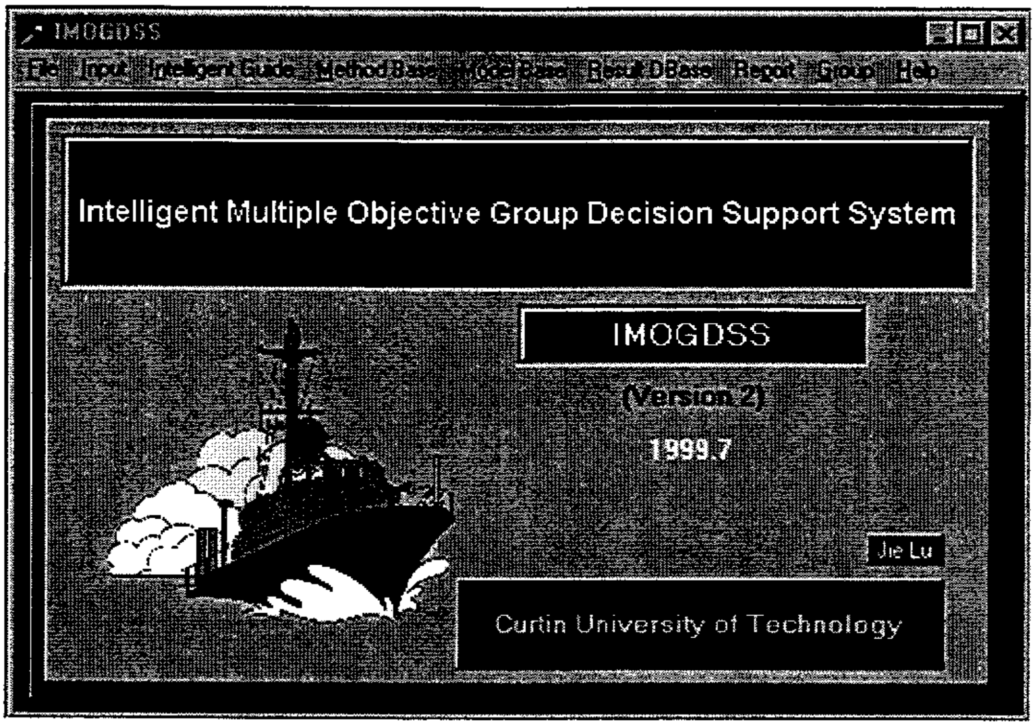


Figure 5-2 System desktop of IMOGDSS

5.3.2 Menu Overview of IMOGDSS

Figure 5-3 shows nine sub-menus. When this IMOGDSS is selected to execute, it immediately assumes that the user wants to start a new application. This New Application item, Open Database File, Open Model File, Print Database File, Print Model File and Exit are included in File sub-menu. Input sub-menu includes Decision Variable Input, Objectives Input, Constraint Input and about input. Intelligent guide sub-menu includes Novice Intelligent Guide, Intermediate Intelligent Guide and Expert. Method sub-menu includes seven MODM methods' names: ESGP, IMOLP, ISGP, LGP, STEM, STEUER, ZW and one liner programming (LP) method. Model sub-menu includes Current User Model and Model Base. Result DBase sub-menu includes Current Data/result and Database. Report sub-menu is utilized to form an individual decision report and a group decision report. Group sum-menu consists of Open solution file, Input solutions, five group aggregation method and Group Solution Show.

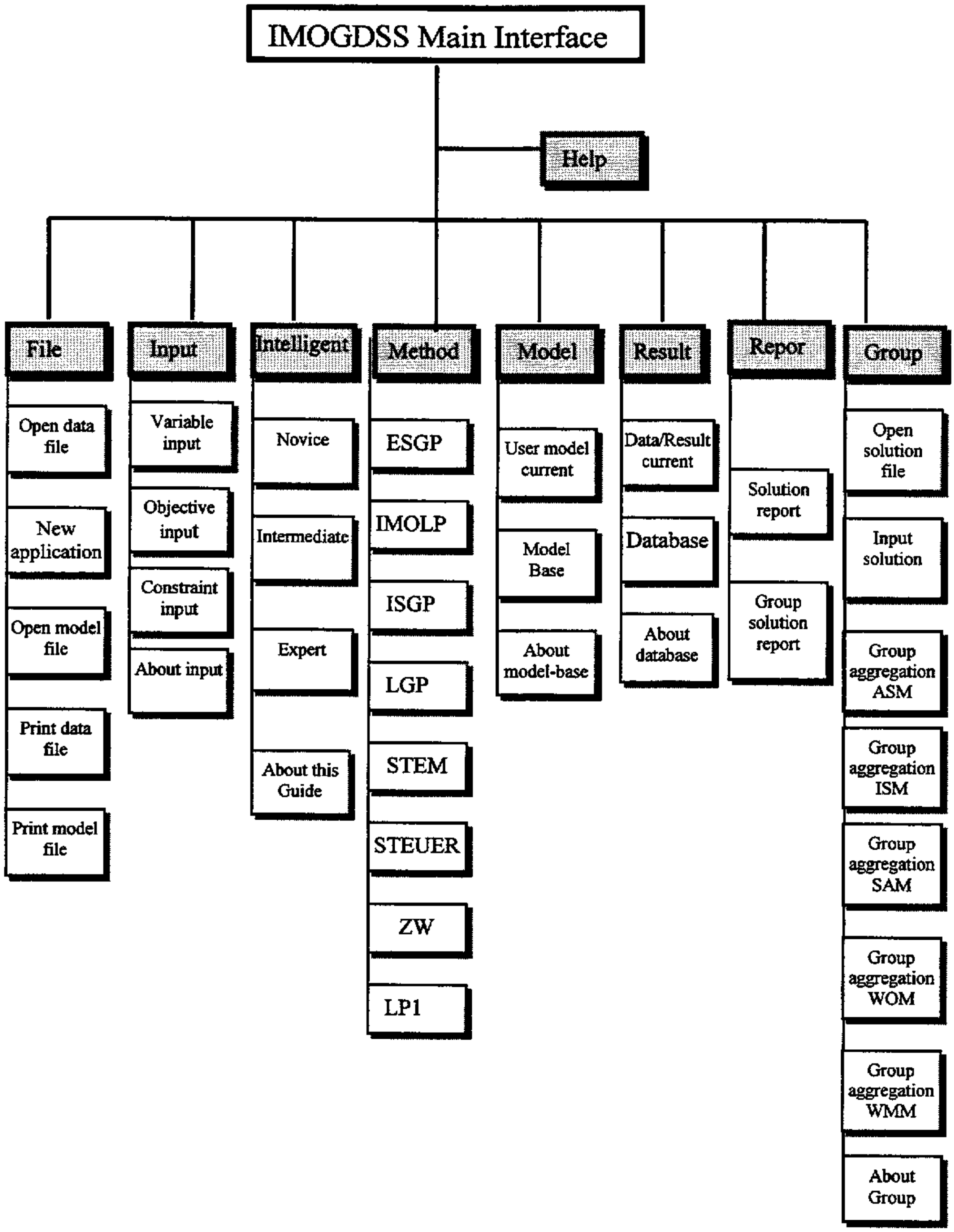


Figure 5-3 System menu overview

5.3.3 Design of Input Subsystem

The input subsystem consists of a set of form-based windows aimed at accepting data of objectives and constraints. The sequence of menus is preset by the system. Decision makers are driven to choose 'decision variables input' menu first (Figure 5-4). The user needs only to input number of decision variables, number of objectives and number of constraints, the variable name will be given automatically such as X(1),...,X(n), F(1), ..., F(m) and C(1), ..., C(k). However, users still have the chance to change these variable names. Figure 5-5 show *Constraint Grid* window, where numerical values of constraint coefficients are entered by filling in a form.

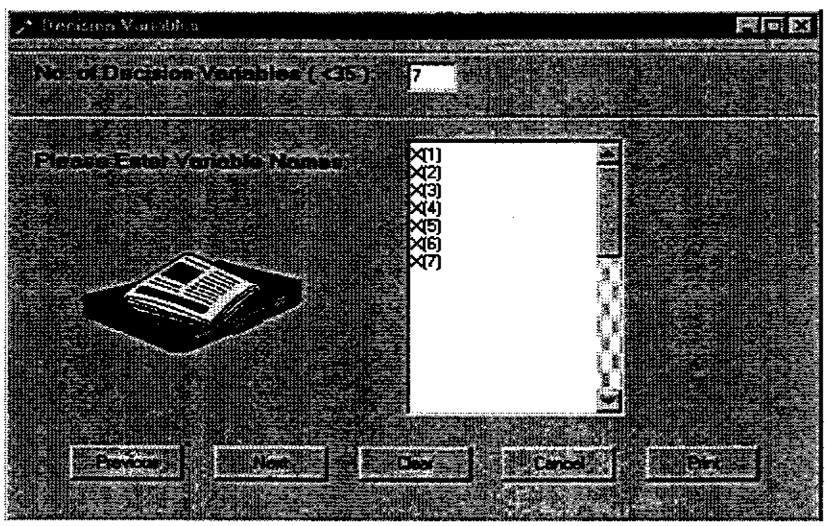


Figure 5-4 Number of decision variables input

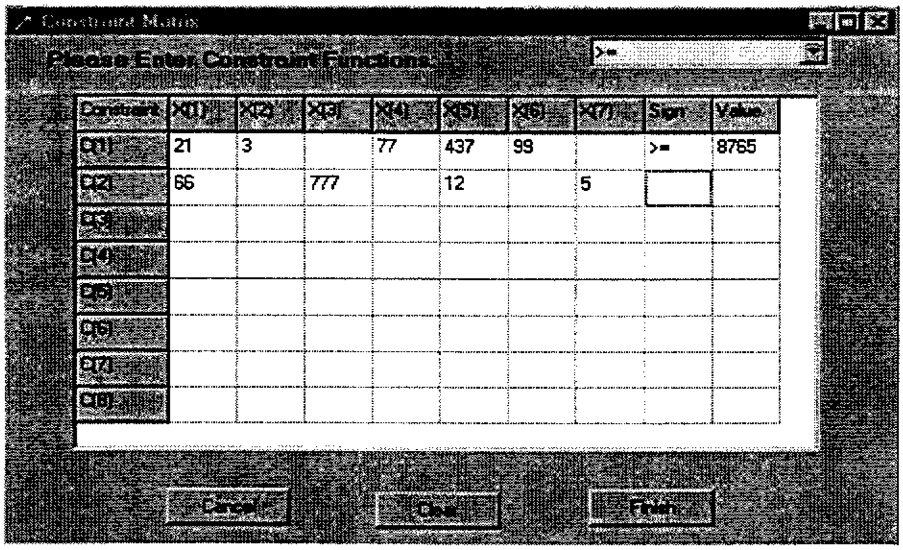


Figure 5-5 Constraint function grid

5.3.4 Report Subsystem

The system report function stores, displays and prints the current individual preferred solution and group decision report using graphics and a form. An individual solution is shown graphically in Figure 5-6. A group of solutions and an 'average' solution are described in Figure 5-7.

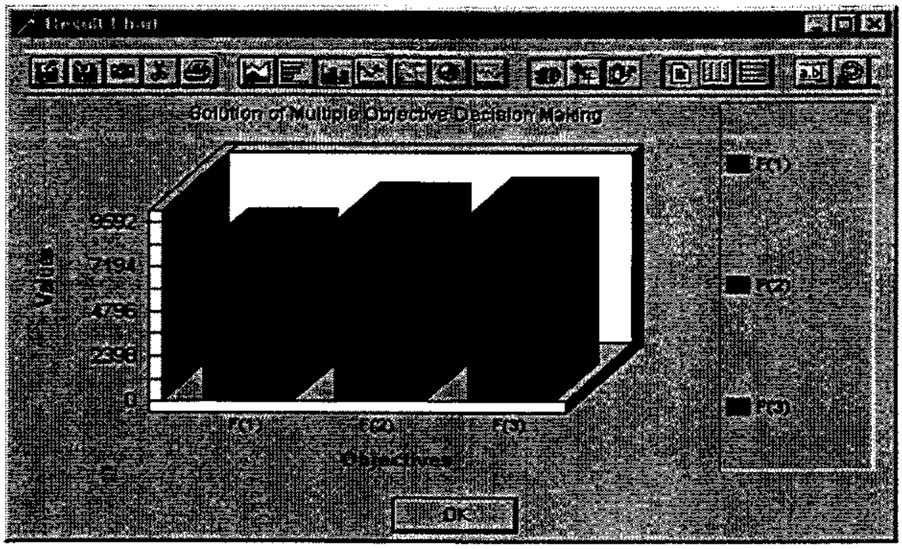


Figure 5-6 A solution is shown by graphical display

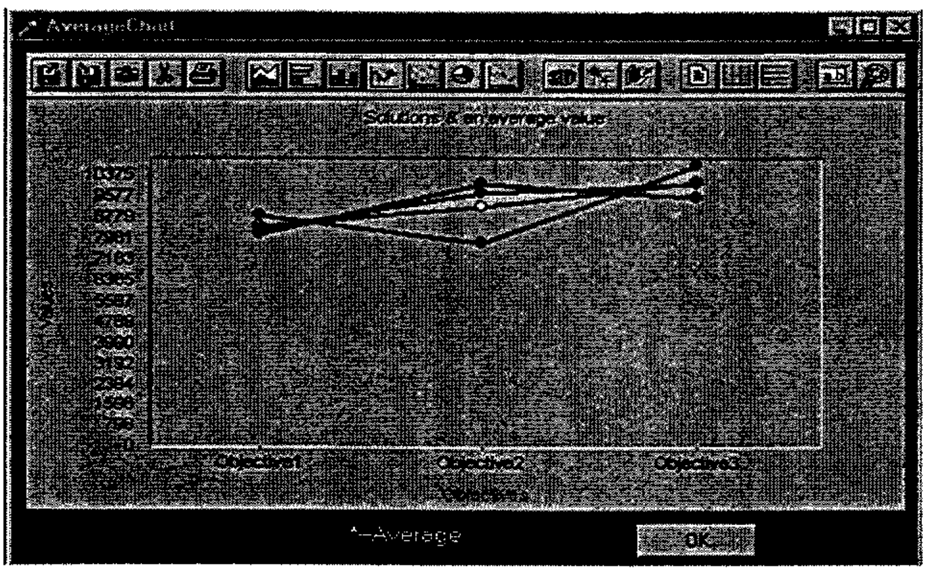


Figure 5-7 A group of solutions and an 'average' solution

5.4 Design of MODM Methodologies Subsystem

A large number of MODM methods have been researched, developed and presented in the literature because of the theoretical challenge and practical applications to a wide variety of decision making problems. Some methods are more suitable and efficient than others in the solution of a particular decision problem for particular decision makers. Hence the IMOGDSS contains a methodology subsystem that is used to execute seven well-established MODM methods available to the decision makers. These methods are, Efficient Solution via Goal Programming (ESGP) (Ignizio 1981), Interactive Multiple Objective Linear Program (IMOLP) (Quaddus & Holzman 1986), Interactive Sequential Goal Programming (ISGP) (Masud & Hwang 1981), Linear Goal Programming (LGP) (Ignizio 1976), Step Method (STEM) (Benayoun et al. 1971), STEUER (Steuer 1977) and Zionts and Wallenius (ZW) (Zionts & Wallenius 1976).

These methods have different characteristics, different requirements for information and information type as well as different required stages (Hwang & Masud 1979). Therefore, they will be suitable for different types of applications and different requirements and knowledge levels of the decision maker. Table 5-1 shows a classification for MODM methods as well as the location of the seven selected methods. The method of no articulation of preference information was not chosen in this MODM methodology base. LGP is a typical method where information is needed in a priori articulation. ZW, ISGP, ESGP, and IMOLP are four interactive methods that provide explicit trade-offs. STEM and STEUER are also interactive methods where information is interacted through implicit trade-off. No method in the last class has been chosen. Because, from the application point of view, we have not found any other methods in this class which have significant defference with the selected methodologies in terms of the requirements for information, information types, and solution process.

Table 5-1 A classification of MODM methods

	Stage at which information is needed	**Type of information**	**Selected methods**
MODM	No Articulation of preference information		
	A priori articulation of preference information	Cardinal information	
		Ordinal & cardinal information	**LGP**
	Progressive articulation of preference information (interactive method)	Explicit trade-off	**ZW**
			ISGP
			ESGP
			IMOLP
		Implicit Trade-off	**STEM**
			STEUER
	A posterior articulation of preference information (non-dominated solutions generation method)	Implicit/explicit trade-off	

IMOGDSS is an interactive system which provides three different kinds of interaction. This point has been discussed in Chapter 3. Table 5-2 shows the situation of seven methods taking three types of interaction. For example, LGP takes a pre-interaction with users before the solution process starts through collecting the weights, goals and priorities of objectives. The IMOLP and ISGP also take a pre-interaction respectively. The method STEM takes a pro-interaction during the solution process. The STEM first displays a solution and the ideal value of each objective to the decision makers. It then asks the decision makers to accept or reject this solution. If the *Accept* button is clicked, then this solution is taken as the final satisfactory solution and the STEM is not executed. However the decision makers have different preferences and different f choices, and they often like to further search so that more solutions are obtained. The *Reject* button is thus clicked and a relaxation process starts. The decision makers are told that they have to accept a certain amount of relaxation of a satisfactory objective to allow an improvement of the unsatisfactory ones in the next cycle. In some cases the decision makers' relaxation values are not available. When the relaxation fails, the system enables the decision makers to continue to re-enter a set of relaxation values. The second solution is found and displayed after the relaxation is completed. The decision makers still can accept or reject this solution. If the decision makers accept it, it is the final satisfactory solution otherwise the system repeats the above-mentioned relaxation process. Post-interaction is used in all methods. After a set of candidate solutions has been generated, the decision makers are required to choose the most satisfactory one. This type of post-interaction can be divided further into 'one-by-one confirmation' and 'choosing confirmation' types.

Table 5-2 Types of interaction of MODM methods

Type Name	**ESGP**	**IMOLP**	**ISGP**	**LGP**	**STEM**	**STEUER**	**ZW**
Pre-interaction		*	*	*			
Pro-interaction	*	*	*		*		*
Post-interaction	*	* *(in some case)*	*	*	*	*	*

It has been found that the seven methods have strong or weak interactionability, including Pre-interaction, Pro-interaction (solution interaction) and Post-interaction. IMOGDSS implemented these abilities so that it has a number of advantages (it needs to be pointed that in some literature, the interactionability only means pro-interaction shown in Table 5-2, this thesis calls the interaction as pro-interaction or solution interaction):

- First, it explores promising solutions rather than finding "the" optimal solution.
- Secondly, it interacts with the problem owner (stake-holder) and thus reflects his/her preference structure.
- Thirdly, it generates alternative solutions for evaluation and selection
- Finally, it becomes a learning process for the decision maker who understands a great deal about the problem.

The result of all these should produce a satisfactory, high quality and high confidence decision for the problems. All seven methods have a common goal i.e. supporting the decision makers by exploring alternative solutions, although their interaction processes are quite different. Therefore, these seven methods are developed as independent executables to facilitate the flexibility required of the system. These methods share similar data acquisition routines and these routines are developed as independent modules so that data acquired could be accessed by all the methods. The method selection function is activated by invoking the method menu. The system produces a list of available methods from which any method can be selected. The system then shows a short description of the selected method. More details of these will be discussed in Chapter 6.

5.5 Design of Knowledge-based Intelligent Guide Subsystem

One of the main advantages of IMOGDSS is the incorporation of an intelligent guide subsystem. The intelligent guide system is built on a knowledge base and is used for:

- identification of the application problem nature and type,
- identification of the decision makers' situation,
- analysis for current information and the decision makers' objective, and
- determination of the best MODM method that matches the needs of the specific problem.

Building a knowledge based intelligent guide system needs knowledge acquisition, accumulation, transfer, and transformation from some knowledge source (Yang 1995). To build the knowledge base of IMOGDSS, the knowledge for the identification of problem and the selection of MODM method(s) is constructed by capturing the basic knowledge about the various methods and their characteristics. Based on the research of Tecle & Duckstein (1992) and Poh (1998), the various characteristics of MODM methods are classified into four classes, there are:

- Decision makers-related,
- Methods-related,
- Problems-related, and
- Solutions-related characteristics.

By studying the characteristics of seven methods implemented in IMOGDSS prototype and obtaining the knowledge from some experts in MODM area about the characteristics of MODM methods, four characteristics analysis models are produced respectively based on the four classes. From the four models, the knowledge about

MODM method characteristics is formed and saved in a knowledge base. These models will be discussed in Chapter 7. Figure 5-8 shows the knowledge capture process.

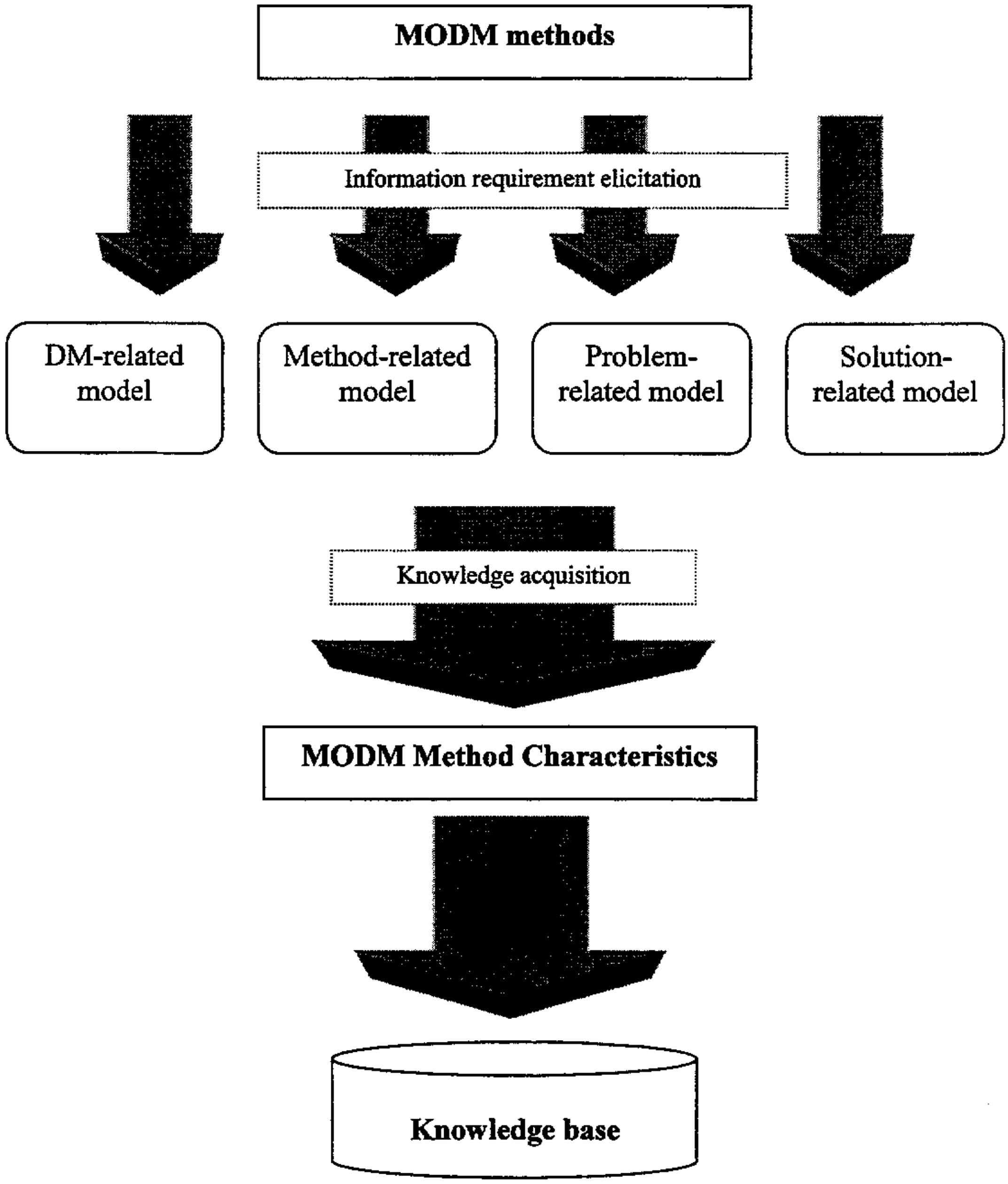


Figure 5-8 Identification and capture of knowledge of MODM method characteristics

To provide appropriate guidance for decision makers possessing different levels of expertise, three modes of guidance have been incorporated namely the intermediate, the novice and expert modes.

The intermediate mode is designed for the decision makers who are familiar with some concepts and methods of MODM, or not so familiar with the methods but do have access to the various inputs. The decision makers can discover which are the methods that correspond to a set of answers by responding to some technical questions based on their problems, their desire for solution and their data preparation.

For the decision makers who are totally unfamiliar with MODM, the novice mode will prompt the decision maker with a set of general non-technical questions regarding the decision problem, expected solutions and the decision makers' preferences. From the answers obtained, suitable methods will be recommended. Figure 5-9 shows the three modes.

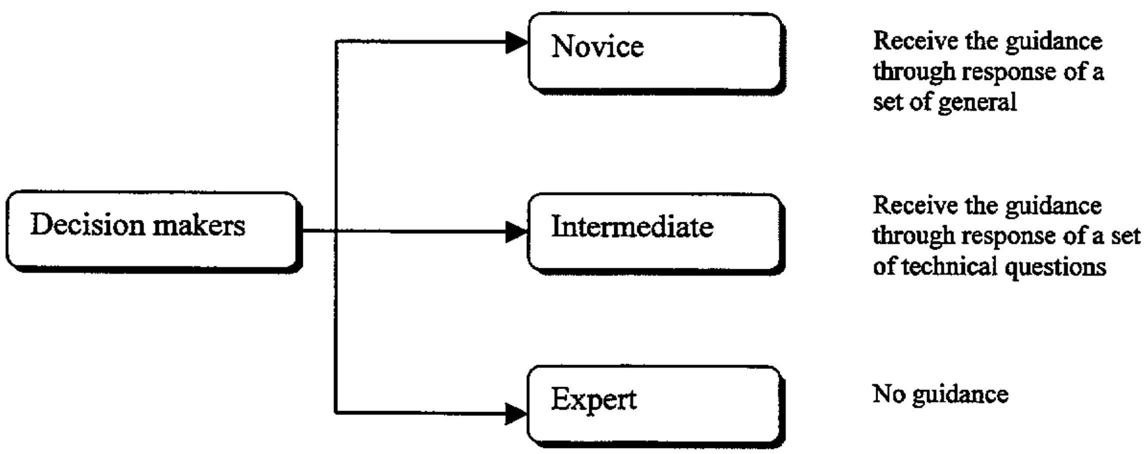


Figure 5-9 Three different level of expertise of decision makers

All questions concerning method selection are shown on serial question-boxes of the graphical interface for Intermediate mode and Novice mode. The decision makers answer each question by clicking each Radio Button. If a decision maker's responses completely match with a method in the methodology base, we call it a completed match, and this method is recommended to the decision maker. However, it is not often that a decision maker's responses exactly match the characteristics of one method. Therefore, a solution that best satisfies (closest) the decision maker's requirement is then found and recommended by missing a characteristic which has the lowest weight.

This subsystem is divided into five components (subsystem) that have different functions (Figure 5-10). The *Question subsystem* asks questions concerning the methods characteristic requirement on serial question boxes of GUI. The *Response subsystem* is to relate the response of each question and to assert them into the knowledge base. The *Ignore characteristic subsystem* is used to find a characteristic which has the lowest weight and ignore it. The *Inference subsystem* is used to reason and solve the method selection problem. The *Method provides subsystem* works to inform the decision maker that a suitable method has been determined.

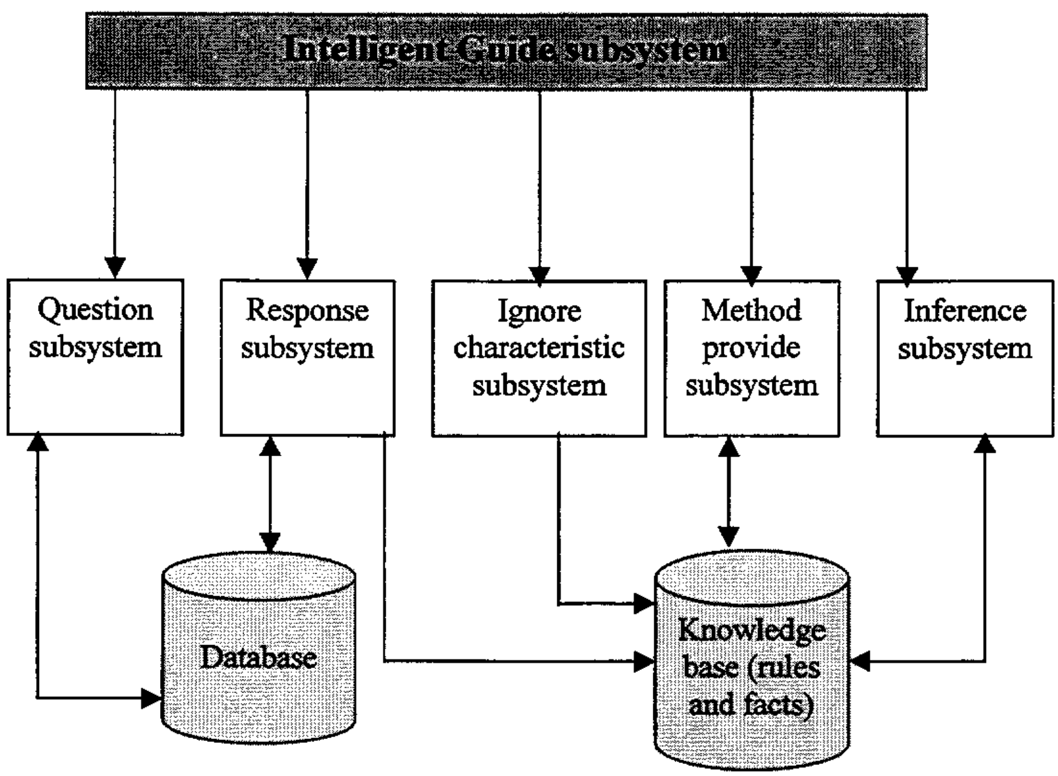


Figure 5-10 System architecture of intelligent guidance subsystem

5.6 Design of Database and Model Base

The data subsystem stores the factual information about the multiple objective problem solving domain in general and the current application in particular. It gets data about objectives and constraints from the input subsystem and solutions of MODM problems from the method subsystem. It also displays solutions and provides the solution data to the *Report subsystem*. Figure 5-11 shows the working process of the database subsystem. In this project, the database is organised as a set of data files for ease of maintenance. The users are allowed to open any data files. Figure 5-12 shows a screen of open data files. Figure 5-13 shows the data of an application.

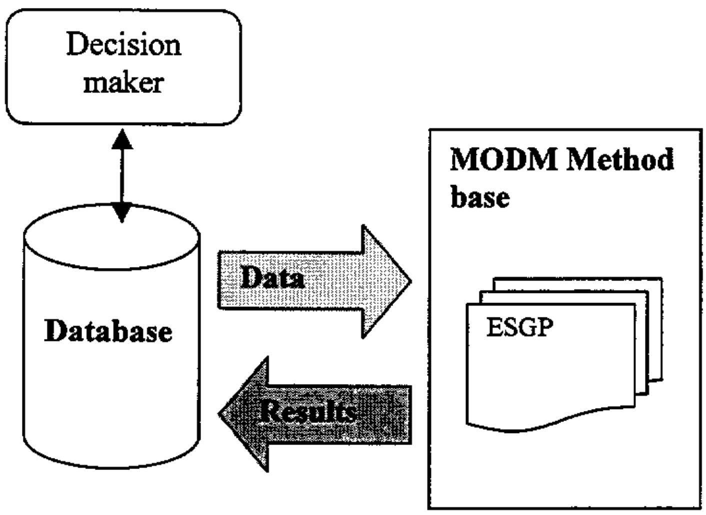


Figure 5-11 The working process of the database subsystem.

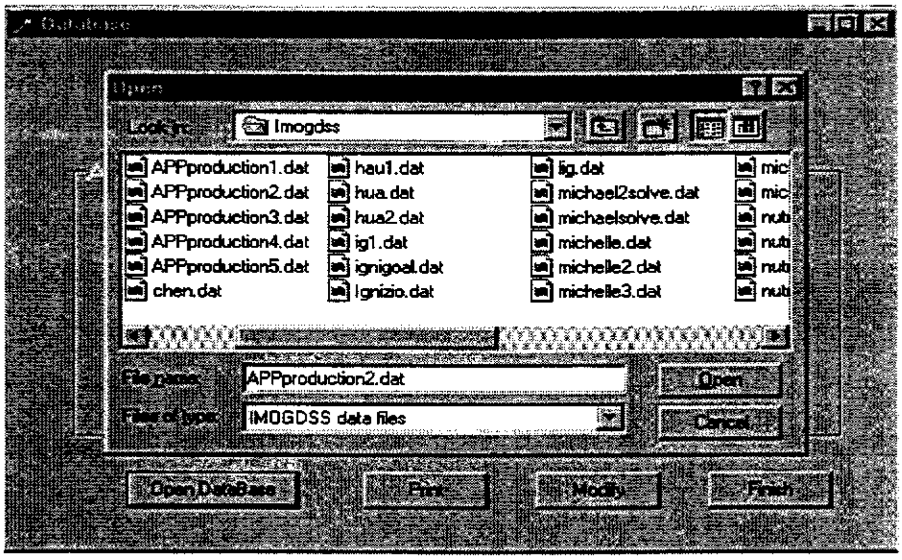


Figure 5-12 Open a data file

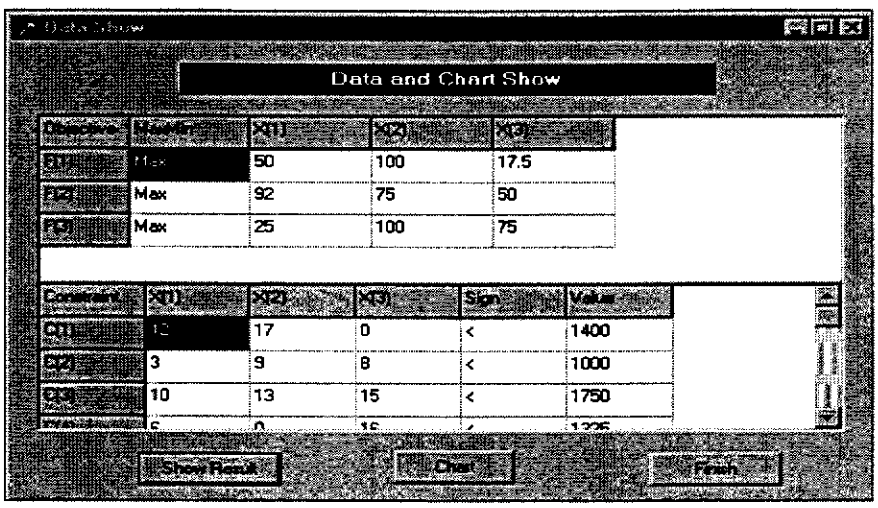


Figure 5-13 A data file show

IMOGDSS is a general tool for solving various MODM problems. Therefore, IMOGDSS has the function of storage the application models. The application model is related to the mathematical model of MODM problems. In fact, each particular application problem (or with its solution) is an application model. The models are developed through an interactive and consultative process between the problem owners and decision makers or decision groups. This solution task is supported by the *methodology subsystem*. Whereas *model subsystem* stores application models before or after the MODM problem is solved. These models can be printed, displayed and modified. The model subsystem can be also used as a mean for the decision maker to determine the goals for objectives through providing an ideal solution. The use of such a mean will help the decision makers to start their exploring process but not influence the decision makers' final decisions, because they still can change their goals for each objective. The model base is easy to maintain where additional models can be added and an existing model can be deleted. Figure 5-14 displays an application model.

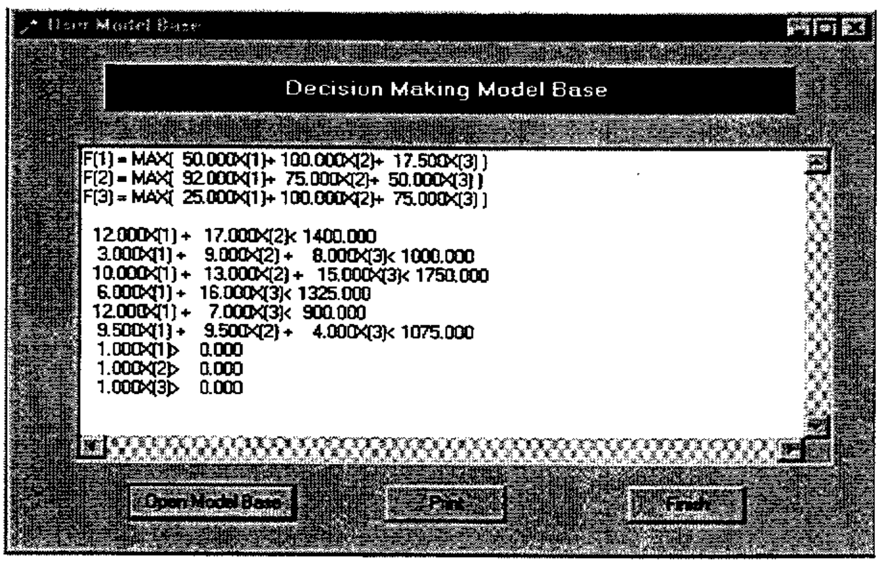


Figure 5-14 An application model is shown

5.7 Design of Group Subsystem

The process of the MODM based GDM process implemented in IMOGDSS is divided into two stages: first, each decision maker makes a decision for an MODM problem; secondly, decision makers negotiate about their decisions so as to achieve a compromise solution of this MODM problem. This group subsystem will implement the second stage.

A decision making group is first formed to exchange information and ideas, and to identify acceptable and desirable solutions. After choosing from among different solutions of the group members provided by any MODM method, the group members take into account of their preferences or wants, which take the form of objective functions. A GDM methodology base that contains five group aggregation methods is used in the second stage. The five GDM methods construct and follow a group decision rule that the best compromise solution has the shortest distance (or weighted shortest distance) to an 'average solution' or to ideal solution, or best satisfactory with group members. The GDM methodology base allows the decision makers to select any aggregation method and allocate their MODM solutions as alternative solutions that are obtained by using different methods on an intelligent framework. Generally the members of the group have conflicting objectives because each member of the group represents a different business function. And not all members of the group have the same information (i.e., goals of the objective functions). All members can work in a decision room or different locations, that is, it can be face-to-face (FTF) communication or computer mediated communication (CMC). The five GDM methods support FTF and CMC respectively. A facilitator is needed in the group. He/she receives all solutions from each member by e-mail, discs or hard copy and enters them into the group subsystem. The facilitator has no influence on the final solution of a group. Once a GDM method is determined an aggregation procedure will start. A group report will be formed once a final group compromised solution is arrived. Figure 5-15 shows the relationship between the group subsystem and other subsystems. More details will be provided in Chapter 8.

A user manual called "Getting Started with IMOGDSS" is presented in Appendix 3.

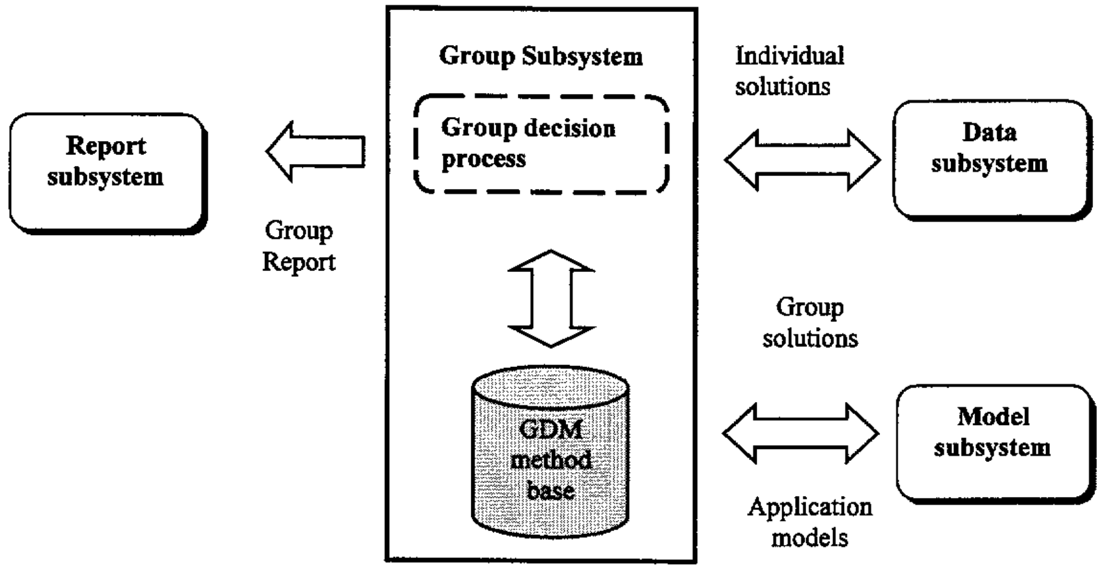


Figure 5-15 The relationship between group subsystem and other subsystems

5.8 Summary

The framework provided in Chapter 4 has been implemented as an integrated set of software tools for operational use by managers in the field. The architecture of the IMOGDSS is proposed in this chapter and some subsystems are discussed. Three main subsystems including the Intelligent guide subsystem, the MODM method subsystem and the Group subsystem will be presented in detail in Chapter 6, 7 and 8 respectively.

The prototype has been implemented in the DELPHI development environment. Its working process can be briefly described as: (1) after the main interface is shown, click the **File** menu to open a new application or an existing application; (2) choose the **Input** menu to enter objective functions and constrains under a nice interface if a new application is opened; (3) click the **Intelligent guidance** menu and go through all questions to obtain a recommendation of the most suitable MODM method based on the current case and a decision maker's preference; (4) click the recommended MODM method's name in the **MODM methods** menu, start a solution process and conduct a solution analysis by following the instruction of the method, and finally obtain a satisfactory solution; (5) the current MODM model and its results can be checked and displayed in Model base and Database respectively; (6) a report can be generated automatically and printed from the **Report** menu; (7) go to the **Group** menu to obtain a compromise solution among all individual solutions using one or more group aggregation methods if this problem needs to be solved in a decision group; (8) a group decision report is produced by choosing the **Report** menu.

The information derived from the software provides answers for the second research question regarding the prototype development, "how an integrated system can be developed effectively consisting of the MODM methodologies, knowledge-based intelligent technology, GDM approaches and DSS framework to support MODM process in both individual and groups environment under an intelligent guidance?"

Chapter 6

KNOWLEDGE MODELLING, KNOWLEDGE BASE BUILDING, AND IMPLEMENTION OF KNOWLEDGE-BASED INTELLIGENT GUIDE SUBSYSTEM[4]

6.1 Introduction

Expert systems (ES) are used extensively in support of managerial decision making. For this reason, ES are frequently referred to as *expert support systems* (Turban 1995). Therefore, it is logical to integrate AI technologies, and especially ES, with a decision support system. Integration of the various management support technologies can be explored best by viewing the support given by such a system. Turban (1995) presented the potential use of seven different expert systems in the integration of ES for the support of a typical managerial process. The major aim of integrated management support is the provision of intelligent capabilities by adding ES (or an intelligent component) to another system.

[4] This chapter is based on papers presented at several conferences:
(1) Lu, J. M.A. Quaddus and R.Williams, " Developing a knowledge-based multi-objective decision making system", *Proceedings of Hawaii International Conference on System Sciences (HICSS)*, 4-7, January, 2000.
(2) Lu, J. M.A. Quaddus and R.Williams, " The design of a knowledge-based guidance system in IMODSS", *Proceedings of Australasian Ccoference on Information Systems,* Wellington, New Zealand, 1-3, December, 1999.

In this study, the purpose of integrating ES to MODSS is to constructing IMOGDSS that provides the intelligent capabilities to enhance the support for MODSS. In IMOGDSS, the ES and MODSS complement each other. Each system performs subtasks that the system is best at. This study provides a new application aspect of ES, that is, utilising knowledge-based ES to select the most efficient MODM method for each decision maker (or decision group) in a particular decision problem. The knowledge-based ES is an intelligent guide to guide a single user or a group of users systematically towards the selection and application of the most appropriate MODM method for their decision making.

This chapter discusses the identification, acquisition, representation and modelling of knowledge, explores the possibility of embedding such a knowledge based intelligent guide within MOGDSS, outlines a specific framework for developing such a knowledge-based intelligent guide, and describes the implementation of the knowledge base and the intelligent guide subsystem.

This chapter provides the answers for the research question 1.2, "How to acquire and express the knowledge about MODM method selection, how to apply the knowledge to guide the decision makers, and how to embed a knowledge-component in this integrated system"?

The knowledge acquisition, representation and modelling are proposed in Section 6.2. Section 6.3 describes the design of knowledge base and integration architecture of knowledge-based intelligent guide with MODSS. Section 6.4 explores the inference process for finding a matched MODM method. The implementation of the intelligent guide subsystem is represented in Section 6.5. An illustrative example for the subsystem application is given in Section 6.6. Finally, a summary is presented in Section 6.7.

6.2 Knowledge Representation and Knowledge-Based Intelligent Guide

6.2.1 Intelligent Guide Embedded in IMOGDSS

A large number of MODM methods have been published in the literature and applied to a wide variety of problems. Some methods are obviously more suitable and efficient than others in the solution of a particular decision problem for particular decision makers. Hence an MODSS or MOGDSS should preferably contain a sufficient number of MODM methods in its methodology base for the decision makers' use.

However, the selection of the most suitable method from such a methodology base is always difficult to accomplish because of the dearth of expertise and experience needed to understand the specific features of the available MODM methods, as well as the ability to match MODM model(s) with current decision needs. Usually only experts in the field are able to take full advantage of the MODSS. This is because sophisticated analytical skills on the part of the decision makers are required to identify the problems and to sequence them according to preferences and match each problem with appropriate MODM methods. Therefore, to utilize the potential of the methodology base effectively, an MODSS should be designed to have the capability of guiding the decision makers to select and use the most suitable MODM methods from the methodology base for solving their decision problems (Bui & Sivasankaran 1988; Pinson & Moraitis 1996; Poh 1998). A knowledge-based intelligent guide is necessary to achieve this aim. To support the selection of methods according to different problem situations and decision makers' situations is an important aspect of an intelligent application in MODSS.

Since organizational decisions are primarily taken in a group environment, a knowledge-based intelligent guide subsystem can be embedded in the group system in order to provide guidance during the whole group decision process. While the intelligent guide subsystem is applied in a group environment, each member can also receive a series of guidance during the solution process based on his/her requirement and each member can accept a recommendation for an appropriate method. Each member gets their satisfactory solutions through using the most suitable MODM method under the most fitting and proper guidance. These solutions then are aggregated into a compromised solution. This compromised solution represents all the members' preferences and expresses the most confident solution for the members of the group. The test results have proved this guide subsystem can produce successful decision aid that explicitly considers the role of preference in the problem solving process in an individual or a group environment. The related results will be discussed in Chapter 9.

6.2.2 Knowledge Representation Embedded in IMOGDSS

Knowledge representation means that knowledge is formalized in a symbolic form. Under a widely used definition, knowledge can be represented as information and posed as different forms in an MODSS. Table 6-1 shows the representational forms of knowledge embedded in an IMODSS. Knowledge can be classified into two main groups: declarative and procedural (Klein & Methlie 1995). Declarative knowledge is a description of facts. It is information about real-world objects and their properties, and the relationships among objects. Procedural knowledge encompasses problem solving strategies, arithmetical and inferential knowledge. Procedural knowledge manipulates declarative knowledge to arrive at new declarative knowledge. A part of the declarative knowledge can be described as the element of a knowledge base, that is 'Facts'. A part of the procedural knowledge can be presented as 'Rules', another element of the knowledge base. After the general discussion for knowledge embedded in IMOGDSS, this chapter will only focus on these two kinds of knowledge representation and the design of a knowledge base.

Table 6-1 Knowledge representation in an IMODSS

	Representation Form	Type	Use	Description
1	Information as data	Declarative knowledge	Describing the objectives, constraints and decision variables	Value of functions & solutions stored in a database
2	Information under the form of text	Declarative knowledge	Describing the meaning of concepts used in the problem formulation	Origin information embedded in a model, form or report
3	Information under a variable name form	Declarative knowledge	Naming variables in a decision model or a decision method	Names of variables and the fields corresponding to the attributes of an entity
4	Structure of MODM methodologies	Procedural knowledge	To provide a list of methods to be used in solving relevant decision problems	Data structures and algorithms
5	Structure of decision models	Procedural knowledge	To provide a list of relevant variables to be used in solving a class of problems	Structure and data model of application problems
6	Structure of reports	Procedural knowledge	To display the information and results of decision making	A report is a complex object mixing text, tables and figures
7	Facts	Declarative knowledge	To describe the methods, characteristics & the relationships between them	Actions based knowledge representation
8	Rules	Procedural knowledge	To describe the inferential strategies for match users' requirements and a method	Rules based knowledge representation

6.2.3 Type of Knowledge-based Guidance

Knowledge can be applied in an intelligent guide system to support various steps in a decision making process, such as information gathering, model development, by different representations. This knowledge-based guidance can be located anywhere in a DSS. Based on Klein & Methlie (1995)' research, six types of knowledge-based guidance can be applied in an MODSS.

(1) Forward Guidance

When a decision maker is using an MODM method, a guidance system can summarise the current step, describe how the current step relates to the previous and next step of the method, and display a status window identifying the current step and the next step. Forward guidance is presented in the form of instructions that clarify the objective of the steps to be performed and explain how the current steps fits into the overall multiple objective decision process.

(2) Backward Guidance

A decision maker can be led backward to complete an unfinished or partially finished previous step of a method. This may be necessary if the user wants to revise the evaluations of each efficient solution. Backward guidance supports feedback, and can be applied when a decision maker is dealing with decision conflict and trade-off during a solution analysis process.

(3) Preventive Guidance

This type of guidance aims to prevent disruption break points, which are miss-understandings of the information and recommendations provided by an MODM method. The focus of preventive guidance is in helping decision makers in interpreting the application of MODM methods.

(4) Rectification Guidance

This type of guidance aims to lead users back to the right way once a wrong way, e.g. an impossible relaxation or an unreasonable goal value, was taken. The rectification guidance is presented in the form of alerts that clarify the failure of the way taken and suggest ways to rectify it.

(5) Compulsion-Receive Guidance

The type of guidance is presented in the form of command that enables an operation to be performed and the user has to select "OK" or a unique enabled button.

(6) Choice-Receive Guidance

In this type of guidance the user can be given more than one way to choose. For example, for a specific decision problem and goals of solution there may be more efficient solutions to match it. The user can select one of them to complete the decision task.

The knowledge-based intelligent guidance will provide the above six types of guidance according to the different problem situations and decision makers' situations to support the selection and use of methods.

6.2.4 Embedded Expert System

Embedded system is one of approaches to physical integration. In MOGDSS, the ES is embedded in a conventional MOGDSS. Embedded system can be considered the 'second generation' of integrating ES and DSS (Turban 1995). IMOGDSS embeds value-added intelligent capability in MODSS to support integrated spreadsheets, text processing, reasoning, graphics, report generation and communication. Users see a single application with which they can work; where no distinction between ES and MODSS exists. Embedded systems, which are usually more efficient than systems with access approaches, could be the most important information technologies of the future. Although embedded systems seem to be desirable, they are more difficult and more expensive to construct (Turban 1995).

6.3. Knowledge Modelling and the Design of Intelligent Component

6.3.1 Acquisition Process of Knowledge on MODM Methodologies

The knowledge acquisition is the process of capturing the expert's knowledge about the domain into a knowledge system. The process includes two main phases: the identification and collection of data, then the representation of the facts representing the expertise to be kept in the system's knowledge base (Klein & Methlie 1995). According to the definitions of Gabriella (1990), the following steps are used to identify and collect the knowledge of MODM methods selection (Figure 6-1).

- ***Method identification:*** identifying a number of traditional and popular MODM methods based on a literature review such as Hwang & Masud (1979), Poh, Quaddus & Chin (1995) to build an MODM methodology base. Chapter 5 has discussed the seven methods' identification processes.

- ***Validity recognition***: a number of validations are recognized. They are conceptual validity, logical validity, experimental validity and operational validity.

- ***Methods comparison***: comparing all methods included in this system through different points of view and classes. The different features of these seven methods were displayed in Chapter 5, Table 5-1.

- ***Characteristics and concepts identification:*** the characteristics and concepts of the MODM methods are identified. These will be discussed in the following sections.

- ***Selection of the type of knowledge representation***: there are four main types of knowledge representation schemes in a knowledge base: production rules, semantic nets, frames and logic. This study used the production rules.

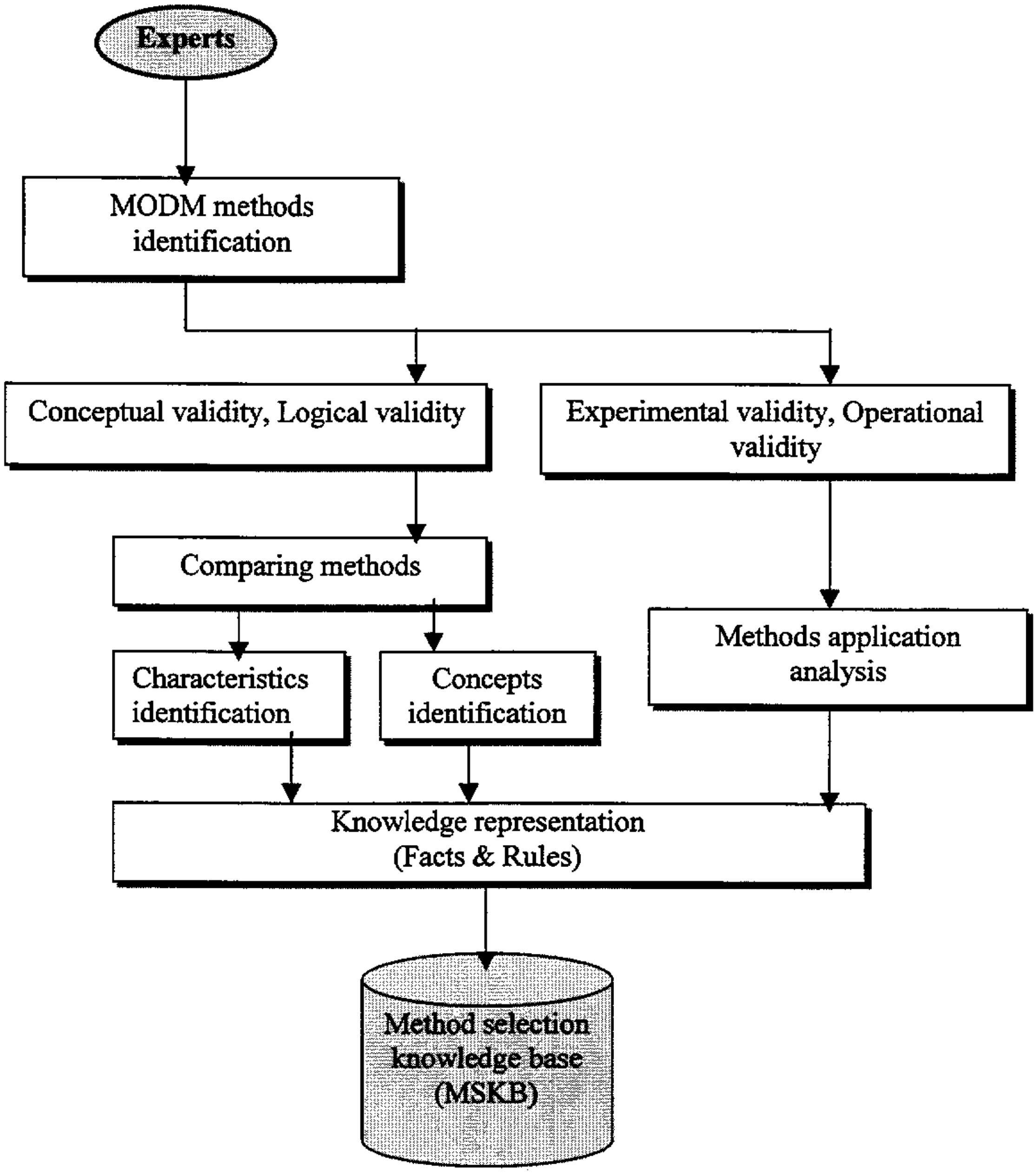


Figure 6-1 Identification of method selection knowledge base (MSKB)

6.3.2 Characteristics Analysis Models of MODM Methodologies

To build the MSKB for the intelligent guide subsystem, the knowledge for the selection of MODM methods is structured by capturing the basic knowledge about the various characteristics of each method and relative method-requirement matching rules. Based on Tecle & Duckstein (1992) and Poh (1998), the various characteristics of MODM methods are identified and collected for four classes, that is Decision Makers-related, Methods-related, Problems-related and Solutions-related characteristics. By studying the characteristics of the seven methods implemented in the IMOGDSS prototype, four characteristics' analysis models are produced respectively in Table 6-2 to Table 6-5. These models identify the characteristics of the seven MODM methods for four classes respectively and will be applied in building the Characteristic-Method Logical Connectivity Models.

- **Decision makers (DMs)-related characteristics analysis model**

DMs-related characteristics are related to the decision makers' preferences for selecting a method to solve a decision-making problem. Some of these characteristics are the decision makers' desire to interact with the system, and the decision makers' ability to provide data for a specific MODM method.

Table 6-2 DMs related characteristics analysis model

N	Categorization of Characteristics	ESGP (1)	IMOLP (2)	ISGP (3)	LGP (4)	STEM (5)	STEUER (6)	ZW (7)
1	DMs' desire to interact with system	H	H	H	L	H	L	H
2	DMs' desire to select a satisfactory solution by themselves	Y	Y	Y	Y	N	Y	Y
3	DMs' desire to analyze the solutions	Y	Y	N	N	Y	N	Y
4	DMs' ability to provide data (e.g. weight, goal)	N	Y	Y	Y	N	N	N

H--high, L--low; Y—yes, N---no.

Table 6-2 shows the DMs-related characteristics of every method. For example, ISGP requires decision makers to have a high desire to interact with the system, to select a satisfactory solution by themselves, and to have the ability to provide necessary data.

- **Method-related characteristics analysis model**

Method-related characteristics are related to the solution process of MODM methods. Some of these characteristics include whether to use a linear programming technique or goal programming, to define or do not define an ideal solution.

Table 6-3 Methods related characteristics analysis model

N	Categorization of Characteristics	ESGP (1)	IMOLP (2)	ISGP (3)	LGP (4)	STEM (5)	STEUER (6)	ZW (7)
1	Uses linear programming technique	N	Y	N	N	Y	Y	Y
2	Uses goal programming Technique	Y	N	Y	Y	N	N	N
3	Defines an ideal solution	Y	N	Y	N	Y	N	N
4	Defines a worst solution	N	N	Y	N	N	N	N
5	Measure of optimality (comparing ideal)	Y	N	Y	N	Y	N	N
6	Uses trade off processing	N	Y	Y	N	Y	N	Y
7	Exploratory (not convergence)	Y	Y	Y	N	Y	Y	N
8	Guaranteed to be efficient solution	Y	Y	Y	N	N	Y	Y

Table 6-3 shows the method-related characteristics of every method. For example, ESGP uses goal programming algorithm and a trade off processing. It also defines an ideal solution and it can guarantee to get an efficient solution for an application.

- **Problems-related characteristics analysis model**

Problem-related characteristics are dependent on the actual decision problem. For example, some MODM methods, such as IMOLP and LGP, require the provision of weights for each objective of a problem by decision makers. ISGP and LGP need users to provide the goals for each objective. IMOLP needs to establish the weights for the objectives in a particular problem and to provide the flexibility to change the weight based on the problem situation. Table 6-4 shows the problem-related characteristics of each method.

Table 6-4 Problems related characteristics analysis model

N	Categorization of Characteristics	ESGP (1)	IMOLP (2)	ISGP (3)	LGP (4)	STEM (5)	STEUER (6)	ZW (7)
1	Establish weights for the objectives in a particular problem	N	Y	N	Y	N	N	N
2	Establish priorities for the objectives in a particular problem	N	N	N	Y	N	N	N
3	Establish the goals for objectives in a particular problem	N	N	Y	Y	N	N	N
4	Change weights for a new cycle	N	Y	N	N	N	N	N
5	Change goals for a new cycle	N	N	Y	N	N	N	N

- **Solutions-related characteristics analysis model**

Solution-related characteristics are related to the types of solution processed. Some MODM methods such as STEUER, ISGP, and LGP produce only a subset of the efficient solutions, while others such as ESGP produce all efficient solutions. For example, it can be found from Table 6-5 that ZW produces a subset of the efficient solutions and decision makers are then required to decrease/increase the value of the objectives in a given solution.

Table 6-5 Solutions related characteristics analysis model

N	Categorization of Characteristics	ESGP (1)	IMOLP (2)	ISGP (3)	LGP (4)	STEM (5)	STEUER (6)	ZW (7)
1	Provides all efficient solutions	Y	N	N	N	N	N	N
2	Provides single solution in a cycle	N	Y	N	Y	Y	N	N
3	Provides a subset of (efficient) solutions in a cycle	Y	Y	Y	Y	N	Y	Y
4	A satisfactory solution selected by DMs	Y	Y	Y	Y	N	Y	Y
5	A satisfactory solution selected by the system (algorithm)	N	Y	N	Y	Y	N	N
6	Decreases/Increases the value of the objectives in a given solution	N	N	N	N	N	N	Y
7	Degrades/Improves the value of the objectives in a given solution	Y	N	N	N	N	N	N
8	Sacrifices/Improves the value of objectives in a given solution	N	Y	N	N	N	N	N
9	Relaxation of objectives in a given solution	N	N	N	N	Y	N	N

6.3.3 Recognition of the Characteristics of MODM Methodologies

The four characteristic analysis models reflect the characteristics of seven MODM methodologies and their different aspects. In order to ensure the consistency of knowledge in a knowledge base, the following operations are applied for combining the characteristics in each characteristic model and to produce the characteristic-method logical connectivity models.

- ***Join***: Every characteristic model displays different features of MODM methods from a different point of view. However, there are duplicated parts in these models. For example, there is characteristic "DMs' desire to select a satisfactory solution by themselves" in Table 6-2, and there is a characteristic "a satisfactory solution selected by DMs" in Table 6-5. This study joined the two characteristics and defined them as a characteristic "D-selected";

- ***Remove***: Some characteristics that do not possess the operational validity, experimental validity or logical validity should be removed. For example, "measure of optimal solution" is not experimentally valid except where an intelligent evaluation system is embedded. However it is contained within the scope of this project. The characteristic "measure of optimal solution" is therefore not considered.

- ***Classify***: Some characteristics are more general and some are very technical. The technical characteristics couldn't be identified by decision makers who have not got enough knowledge of MODM methods for selecting one of these methods. So, as mentioned in Chapter 5, the characteristics must be captured into two groups in order to build the guidance modes as a front-end of the knowledge-base. One group of characteristics is used to serve "Novice" users by a set of non-technical question boxes and another is used to serve "Intermediate" users by a set of technical question boxes.

6.3.4 Novice and Intermediate Modes for Intelligent Guide

The novice mode consists of non-technical characteristics that are applied to the decision makers who are totally unfamiliar with MODM. The novice mode will correspond to a set of general non-technical questions regarding the decision problem, the expected solution(s) and the decision makers' preferences. Based on the responses of users, the most suitable method will be found and recommended by running the knowledge base. A total of 10 characteristics are identified for the novice mode (Table 6-6).

Table 6-6 Characteristics (Char.) and facts related to the novice mode

Char. No.	**Char. Name**	**Characteristic Definition**	**Char. Facts**
1	Interaction	more interaction with the system	Char1
2	Subset	system provides a set of solutions	Char2
3	Unique	system provides a unique solution	Char3
4	S-selection	select one satisfactory solution by system	Char4
5	D-selection	select one satisfactory solution by DM	Char5
6	Analyse	analyse the solutions (e.g. improving/ sacrificing the value of objectives)	Char6
7	Ideal	system defines an ideal solution	Char7
8	Weight	prepare the weight for every objective	Char8
9	Goal	prepare the goal for every objective	Char9
10	Priority	prepare the priority for every objective	Char10

The intermediate mode is designed for the decision makers who are familiar with some concepts and methods of MODM, or not so familiar with the methods but have basic knowledge to data analysis during the solution process. The model captures 14 characteristics of MODM methods (Table 6-7).

Table 6-7 Characteristics and facts related for intermediate mode

Char. No.	**Char. Name**	**Characteristic Definition**	**Char. Facts**
1	Interaction	more interaction with the system	Char1
2	Subset	system provides a set of solutions	Char2
3	Unique	system provides a unique solution	Char3
4	S-selection	select one satisfactory solution by system	Char4
5	D-selection	select one satisfactory solution by DM	Char5
6	Trade-off	use a trade off process (more than one cycle) to get a satisfactory solution	Char6
7	Relaxation	use a relaxation method to get a satisfactory solution by improving/sacrificing or decreasing/increasing the value of objectives	Char7
8	Ideal	the system defines an ideal solution	Char8
9	Worst	the system defines a worst solution	Char9
10	Efficient	view all the efficient solutions	Char10
11	Converge	the method to converge to a final solution or just to explore in the feasible region	Char11
12	Weight	prepare weight for every objective	Char12
13	Goal	prepare goal for every objective	Char13
14	Priority	prepare priority for every objective	Char14

6.3.5 Characteristics-Method Logical Connectivity Models

Bui & Sivasankaran (1988) discussed 4 MADM methods for matching their 9 assertions. Poh (1998) identified the relationship between 17 MADM methods (some of them were not implemented) and their 19 characteristics. In this thesis, 7 MODM methods included in IMOGDSS are thoroughly studied and classified according to one or more of the 10 characteristics for the novice mode and 14 characteristics for the intermediate mode. Figure 6-2 shows the logical connectivity between the MODM methods and the 10 characteristics for the novice mode. As an example as shown in Figure 6-2, the ISGP method is characterized by the characteristics of the 'Interaction', 'Subset', 'D-selection', 'Ideal' and 'Goal'. Figure 6-3 shows the logical connectivity between seven methods and 14 characteristics for the intermediate mode.

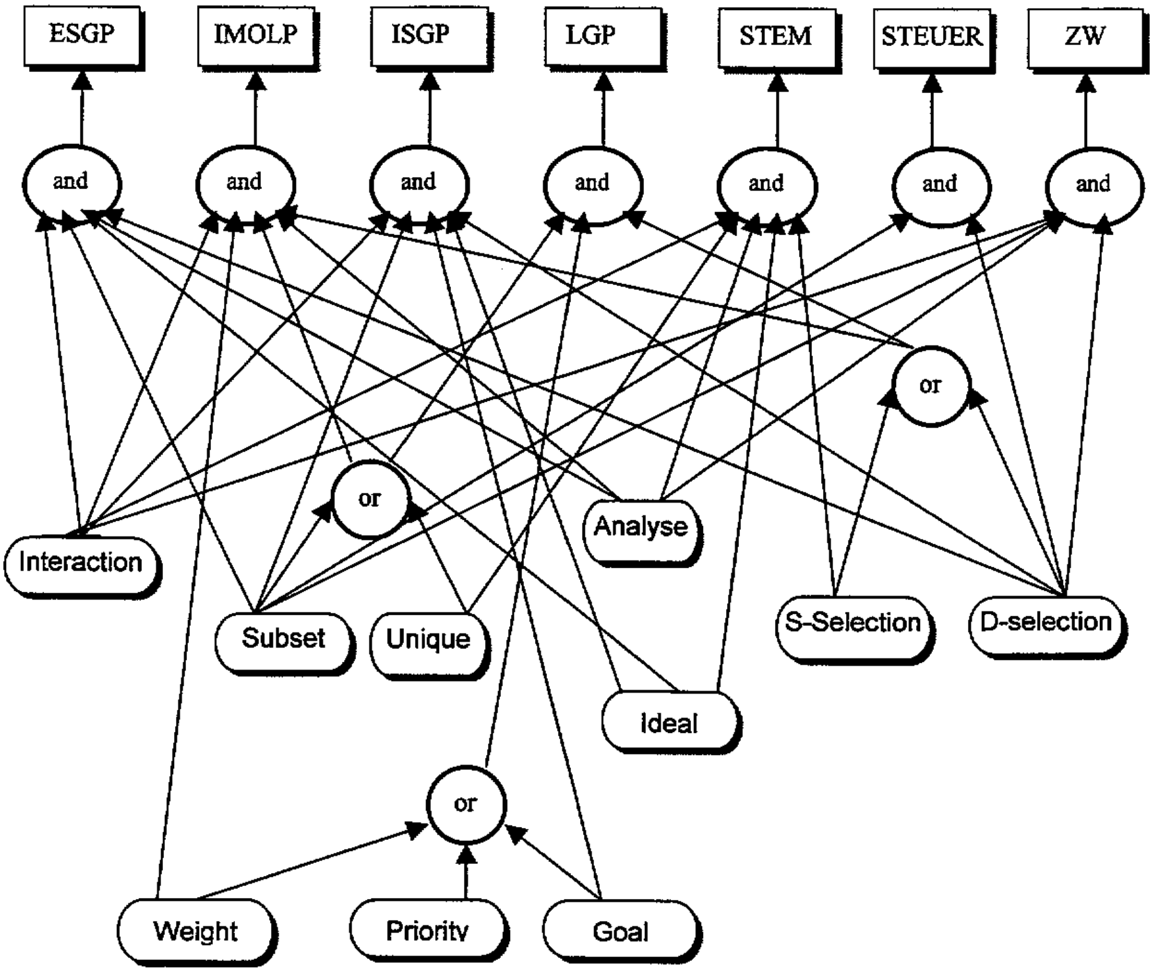


Figure 6-2 Characteristic-method logical connectivity model for novice mode

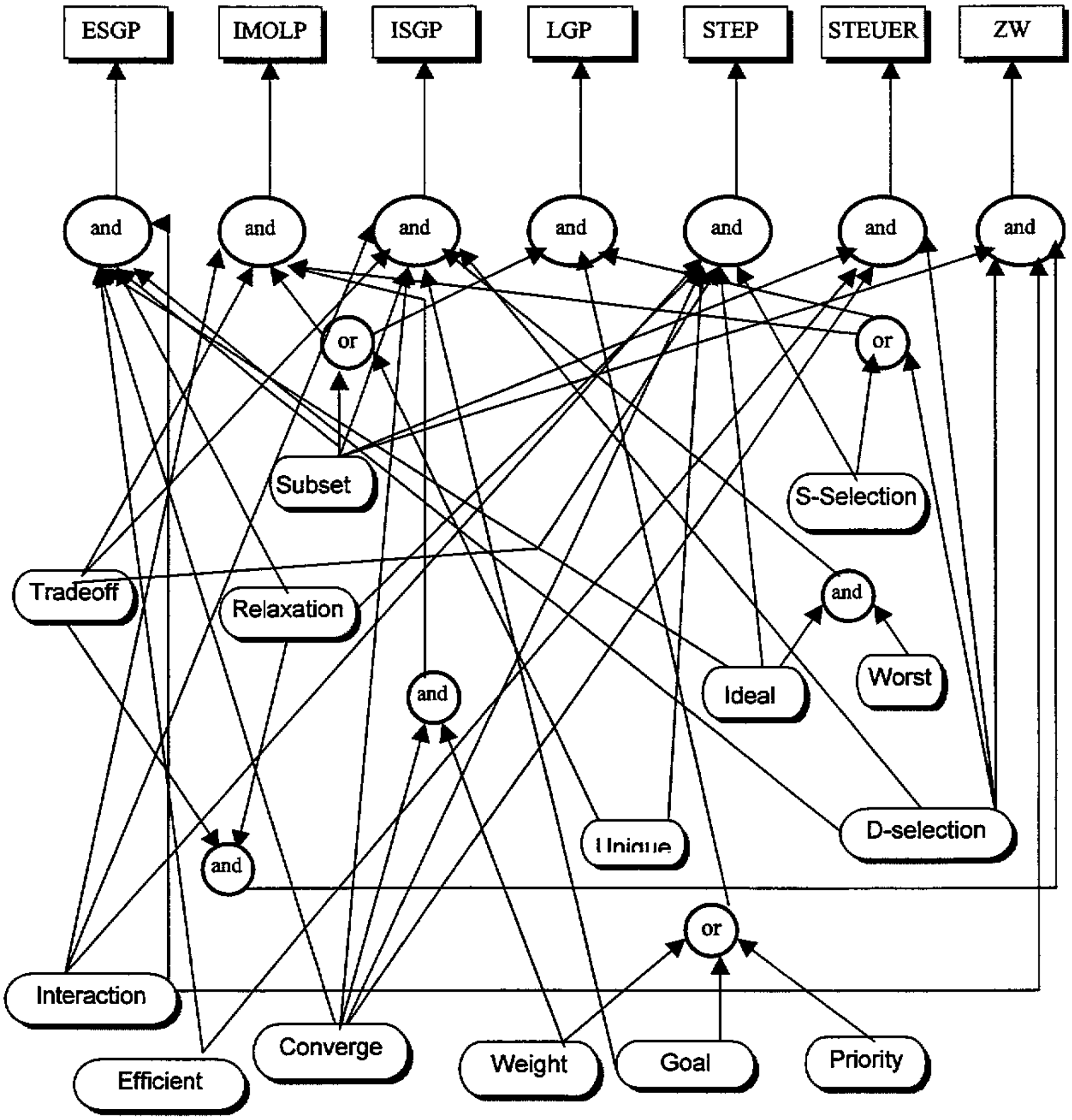


Figure 6-3 Characteristic-method logical connectivity model for intermediate mode

6.3.6 Performance Modelling: Questions and Responses

To obtain the situation of users and the preference of users for an MODM method, two groups of questions are designed and shown to the two levels of decision makers through a series of question dialog boxes. Every question dialog box includes one question with two response items (Radio buttons): Top (T) and Bottom (B) and a set of weights for the question to choose. The decision makers answer each question by clicking each Radio button. All questions concerning method selection are shown on serial question-boxes of a DELPHI interface. The decision makers should also choose one of weights to indicate the degree of importance of this question at the same time as answering each question. The decision makers then can go down to the next question. He/She can also go back to the previous question to change the response, or exit the question dialog box at any question to take the default response. These responses will be converted to "Fact" and are used by inference engine to match the characteristics of one method, and the marked weights are used to measure which method is the most appropriate if no completed method matches with the decision maker's preference. The relationships between questions, responses and characteristics are shown in Table 6-8 for the novice mode and Table 6-9 for intermediate mode.

Table 6-8 Question-answer-characteristic for novice mode

No	Questions	Responses	Char. Name	Char. No
Q1	Would you like to have more interaction with the system?	T	Interaction	1
		B	No	Not
Q2	Would you like the system to provide a set of solutions or a unique solution?	T	Subset	2
		B	Unique	3
Q3	Would you like the system to select one satisfactory solution or would you like to select a solution?	T	S-selection	4
		B	D-selection	5
Q4	Would you like to analyse solutions (improving/sacrificing the value of objectives)?	T	Analyse	6
		B	No	Not
Q5	Would you like the system to define an ideal solution?	T	Ideal	7
		B	No	Not
Q6	Have you prepared a weight for every objective?	T	Weight	8
		B	No	Not
Q7	Have you prepared a goal for every objective?	T	Goal	9
		B	No	Not
Q8	Have you prepared a priority for every objective?	T	Priority	10
		B	No	Not

Table 6-9 Question-answer-characteristic for intermediate mode

No	Questions	Responses	Char. Name	Char. No
Q1	Would you like to have more interaction with the system?	T	Interaction	1
		B	No	Not
Q2	Would you like the system to provide a set of solutions or a unique solution?	T	Subset	2
		B	Unique	3
Q3	Would you like the system to select one satisfactory solution or to select a solution yourself?	T	S-selection	4
		B	D-selection	5
Q4	Would you like to use a trade off process to get a satisfactory solution (more than one cycle)?	T	Trade-off	6
		B	No	Not
Q5	Would you like to use a relaxation method (by improving/sacrificing the value of objectives in a given solution) to get a satisfactory solution?	T	Relaxation	7
		B	No	Not
Q6	Would you like the system to define an ideal solution?	T	Ideal	8
		B	No	Not
Q7	Would you like the system to define a worst solution?	T	Worst	9
		B	No	Not
Q8	Would you like to view all the efficient solutions?	T	Efficient	10
		B	No	Not

Q9	Would you like the method to converge to a final solution or just to explore in the feasible region?	T	Converge	11
		B	No	Not
Q10	Have you prepared a weight for every objective?	T	Weight	12
		B	No	Not
Q11	Have you prepared a goal for every objective?	T	Goal	13
		B	No	Not
Q12	Have you prepared a priority for every objective?	T	Priority	14
		B	No	Not

6.4 Match Strategy, Inference Process and Verification of Rule Base

This section discusses the inference process for finding a matched MODM method in Novice mode. The process for the intermediate mode is similar.

6.4.1 Match Strategy: Completed Match and *n*-Step Match

Let $M = \{M_1, M_2,..., M_7\}$ be a methodology set, $\overline{C}$ be a characteristics set of MODM methods, $C_i = (C_{i1}, C_{i2},..., C_{ik})$ and $C_{ij} \in \overline{C}$ (j=1,2,..,k) be characteristics of M_i, $R = (R_1, R_2,.., R_k)$ and R_i (i=1,2,..,k) be characteristics of the decision makers preferences (it is covered by the responses of the decision makers for the questions) and for any $p \in \{1,2,...,k\}$ there exists an i and j such that $C_{ij} = R_p$, $W = (W_1, W_2,.., W_k)$ be a weight vector for R, k =10 for novice mode and k =14 for intermediate mode.

Definition 1. RC Completed match

If there exists i∈{1,2,..,7} such that for any j∈{1,2,..,k},

$$R_j = C_{ij},$$

we then say R and M_i is a RC completed match and denote it as $R \equiv C_i$, *or* $R^0 = C^0{}_i$. A completed match means the characteristics of a method completely match with the decision makers preferred characteristics.

Definition 2. RC n-step match

Let $R^n = (R_{j_1}, R_{j_2},.., R_{j_{k-n}})$, and

$\{R_{j_1}, R_{j_2}, \cdots, R_{j_{k-n}}\} \subset \{R_1, R_2, \cdots, R_k\}, n = 1, 2, \cdots, k-1$. If there exists i∈{1,2,…,7} such that $\forall\ j \in \{j_1, j_2,.., j_{k-n}\}$

$$R_j^n = C_{ij},$$

we then say R and M_i is a RC n-step match and denote it as $R^n = C_i^n$, and n is called a match degree, where $C_i^n = (C_{ij_1}, C_{ij_2}, ..., C_{ij_{k-n}})$ and $C_i^n = \{C_{ij_1}, C_{ij_2}, ..., C_{ij_{k-n}}\} \subset \{C_{i1}, C_{i2}, .., C_{ik}\}$. An n-step match means that only k-n characteristics of a method match with the decision makers preferred characteristics.

Theorem 1. If for any $i \in \{1,2,..,7\}$, and R and M_i is not a completed match, then there exists n<k, such that R and M_i is RC n-step match.

Proof.

(1) If there exists $k>m\geq 1$ such that $w_{i_j} = \min\{w_1, w_2, ..., w_k\}, j = 1,2,...m$, and since $\{i_1, i_2, ..., i_m\}$ is an order set, we can take j_0 such that j_0 is the least element of $\{i_1, i_2, ..., i_m\}$.

If $i_{j_0} = 1$, we take

$$R^1 = (R_2, \cdots, R_k)$$
$$C_i^1 = (C_{i2}, \cdots, C_{ik})$$
$$W^1 = (W_2, \cdots, W_k)$$

If there exists $i \in \{1,2,..,7\}$, such that

$$R^1 = C_i^1$$

Then R and M_i is 1-step match. If 1< i_{j_0} <k, we take

$$R^1 = (R_1, \cdots, R_{ij_0-1}, R_{ij_0+1}, \cdots, R_k)$$
$$C_i^1 = (C_{i1}, \cdots, C_{ij_0-1}, C_{ij_0+1}, \cdots, C_{ik})$$
$$W^1 = (W_1, \cdots, W_{ij_0-1}, W_{ij_0+1}, \cdots, W_k)$$

If there exists $i \in \{1,2,..,7\}$, such that

$$R^1 = C_i^1$$

Then R and M_i is 1-step match. If $i_{j_0} = k$, we take

$$R^1 = (R_1, \cdots, R_{k-1})$$

$$C_i^1 = (C_{i1}, \cdots, C_{ik-1})$$

$$W^1 = (W_1, \cdots, W_{k-1})$$

If there exists $i \in \{1,2,..,7\}$, such that

$$R^1 = C_i^1$$

Then R and M_i is 1-step match.

(2) If $w_1 = w_2 = ... = w_k$, we take

$$R^1 = (R_2, ..., R_k)$$
$$C_i^1 = (C_{i2}, ..., C_{ik})$$
$$W^1 = (W_2, ..., W_k)$$

If there exists $i \in \{1,2,..,7\}$, such that

$$R^1 = C_i^1$$

Then R and M_i is 1-step match.

(3) If R and M_i is not a 1-step match, we replace R, C_i *and* W by R^1, C^1 and W^1, and repeat the above-mentioned process. If R and M_i is a 2-step match, we have finished the proof of this theorem. If R and M_i is not a 2-step match, we replace R^1, C_i^1 *and* W^1 by R^2, C_i^2 *and* W^2 and repeat the above process again. Finally, as

for any $p \in \{1,2,...,k\}$ there exists an i and j such that $C_{ij} = R_p$, there exists $n<k$ such that R and M_i is an n-step match.

6.4.2 Ignoring Characteristic Match Strategy (ICMS) Based Inference

Decision makers have different problems, different knowledge backgrounds and different preferences for decision making. Their different choices for each question response and intensity of importance of these choices are obtained by using a set of question dialog boxes and weight boxes. The responses and weights are converted to a response vector R that consists of the characteristics the decision maker needs, and a weight vector W that consists of the weight of each characteristic. If a decision maker's responses are a *RC* completed match with a method, this method is recommended and the ICMS is not executed. However, it is not often that a decision maker's responses exactly match the characteristics of one method (that is an RC completed match). An ICMS is thus used based on theorem 1 to find M_{i0} such that a RC n-step is found. The objective of this method is to combine the decision makers' preferences and the weights for each characteristic to find the most suitable method that best satisfies the decision maker's requirement.

The algorithm is centred around an ignoring process based upon a weight vector W, whose elements represent the intensity of the importance of the characteristics that the decision makers prefer. Through this weight vector, the lowest weight element W_l ($1 \leq l \leq k$) is obtained from weight vector W. R_l and C_{il} (i=1,...,7) that correspond to W_l are then found and ignored, if for any *i, R* and *Ci* is not a completed match. If there is an existing M_i such that R and C_i is a 1-step match, this method M_i is then recommended to the decision makers. Otherwise, the second lowest weight is determined, another characteristic is missed and a *RC* 2-step match is measured. Based on theorem 1, an n-step match method will be found after ignoring process *(n<k)* n times. Figure 6-4 shows the ignoring process. The ICMS method is expected to cause the least regret among the decision makers and the greatest comfort and the best coefficient for the method selection process.

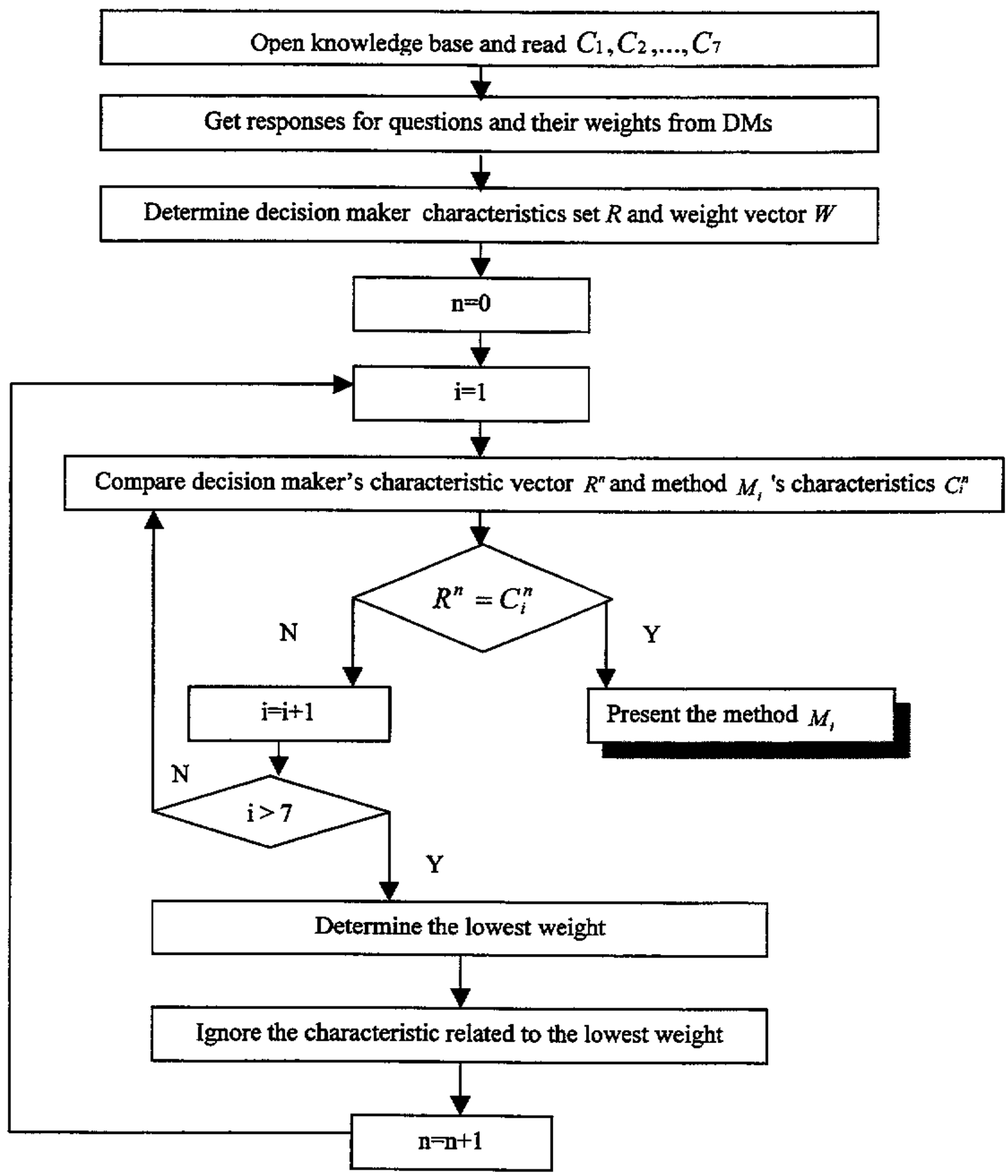


Figure 6-4 Ignoring characteristic process for *n*-step match

6.4.3 Weight Vector (WV) Based Inference

6.4.3.1 Building the Users Weight Vector

Each characteristic is given an intensity of importance by the decision maker. A weight vector of characteristics is therefore built. The lowest weight is then obtained by ranking this weight vector. The characteristic that corresponds to the lowest weight is considered to be ignored first if no method completely matches *R*. The characteristic corresponding to the second lowest weight is missed if no method is found to meet the 1-step match. According to this methodology, two different methods may be recommended to two different decision makers because they are assigned different weights for characteristics even though their responses for the questions were the same.

Four levels of the weights are thus defined. The decision makers can choose any one from each weight dialog box. The choices are:

- Very important,
- Important,
- General,
- Less important.

The decision makers' assignment for the weight can generate a weight vector

W=$(W_1, W_2, \ldots, W_k)$, where

$$W_i = \begin{cases} 2 & if \quad R_i \quad is \quad less \quad impt. \\ 4 & if \quad R_i \quad is \quad general \\ 6 & if \quad R_i \quad is \quad impt. \\ 8 & if \quad R_i \quad is \quad very \quad impt. \end{cases}$$

impt.—important.

Based on theorem 1 the set of weights can be ranked and for any $i<j$, when $W_i = W_j$, W_i is first taken.

6.4.3.2 Defining the System Weight Vector

This system allows the decision makers to default the weight assignment. In this case the system will provide a weight vector W following a heuristic method. This weight vector W is defined as follows:

$$W = (W_1, W_2, ..., W_k)$$

$$W_i = (P_i + Q_i)/2, i = 1,2,.., k,$$

where $P_i \in P$ is a relative coefficient of the characteristic set $C_i \subseteq \overline{C} = C_1 \cup C_2 \cup ...C_7$, Q_i is a changeable coefficient for $R_i, i = 1,2,\cdots k$.

The vector P is obtained through analysing the degree of relation between characteristics C_i and M_i, $i=1,2,..,7$ in Table 6-10. For example, Char1 is related to five methods, so P_1=5. If a characteristic has a higher relative coefficient then it should have a higher weight. We have obtained

$$P = (P_1, P_2, ..., P_{10}) = (5,6,3,3,6,4,3,2,2,1).$$

Table 6-10 Characteristics-method matrix for the novice mode

Char Fact	ESGP	IMOLP	ISGP	LGP	STEM	STEUER	ZW	Pi
Char1	●	●	●		●		●	5
Char2	●	●	●	●		●	●	6
Char3		●		●	●			3
Char4		●		●	●			3
Char5	●	●	●	●		●	●	6
Char6	●	●			●		●	4
Char7	●		●		●			3
Char8		●		●				2
Char9			●	●				2
Char10				●				1

Ignoring a characteristic means changing a request of the users. Some changes in the decision makers' request for characteristics can be easily accepted by the decision makers, others are harder to accept. The acceptability for changes is thus considered. The objective is to produce the least influence upon the decision makers' selection process when a characteristic is ignored. This study defines

E -- easy to accept the changes;

L -- less hard to accept the changes;

H -- hard to accept the changes;

VH -- very hard to accept the changes,

and the scores of E, L, H and VH are 2, 4, 6 and 8 respectively. Based the identification for each characteristic by experts and analysis of test process, a " acceptability vector" is generated as follows:

$$Q = (Q_1, Q_2, ..., Q_{10}) = (8,6,6,6,6,4,4,2,2,2).$$

If a characteristic has hard acceptability for changes for a decision maker then it should have a high weight. Therefore, the W vector is generated.

$$W=(6.5, 6, 4.5, 4.5, 6, 4, 3.5, 2, 2, 1.5).$$

In this case the characteristic 10 in relation to the least weight (1.5) will be ignored first when a failed search match happened, while characteristic 1 in relation to weight (6.5) will not be missed during the whole searching process. Table 6-11 show the vectors P, Q, W and the ignore (missing) ranking for the Novice mode. Similarly, a weight vector W can be generated for Intermediate mode.

Table 6-11 Weights and ignoring ranking for the novice mode

Char NO	Char Name(C)	P	Q	Q	W	Ignore Ranking (N)
1	Interaction	5	VH	8	6.5	(10)
2	Subset	6	E	6	6	(8)
3	Unique	3	E	6	4.5	(6)
4	S-selection	3	E	6	4.5	(7)
5	D-selection	6	E	6	6	(9)
6	Analyse	4	A	4	4	(5)
7	Ideal	3	A	4	3.5	(4)
8	Weight	2	H	2	2	(2)
9	Goal	2	H	2	2	(3)
10	Priority	1	H	2	1.5	(1)

6.5. Design of Knowledge Base and Implementation of the Intelligent Guide Subsystem

6.5.1 Composition of Intelligent Guide Subsystem

AI is concerned with designing intelligent computer systems that exhibit the characteristics associated with intelligence in human behaviour, e.g., decision problem solving. An ES is an AI program designed to provide, or to aid in the determination of, and solutions of expert-level quality to complex problems. This approach to problem solving emphasizes the role of expert knowledge, which is often experiential and more easily represented symbolically, rather than the role of scientific theory, which is most often described mathematically and/or algorithmically. An ES tries to emulate the problem solving behaviour of an expert in his or her area of expertise.

There are two main parts of an ES --- the knowledge base and the inference engine. In addition, subsystems typically exist to facilitate interaction with users, to provide an explanation capability for proposed alternatives, and to acquire knowledge for the knowledge base.

The intelligent guide subsystem in the IMOGDSS consists of the question subsystem, response subsystem, method-show subsystem, Ignoring (missing) characteristic strategy (ICS) subsystem, main-control subsystem and a knowledge base system that includes a knowledge base (MSKB) and an inference engine provided by the expert system shell CLIPS. MSKB includes a set of Facts and a set of Rules.

The question subsystem first questions decision makers by an elicitation technique. The responses are received by the response subsystem. The responses to each question are asserted in the working memory by the inference engine, and responses to the weight of each question are sent to the ICS subsystem. If a suitable method(s) is found the name of method(s) will be displayed to the decision makers at the end of the inference process by the method-show subsystem, else a fuzzy (*n*-step) matched method with the lowest match degree *n* is provided through related facts as asserted by the ICS subsystem.

6.5.2 Design of Facts

Facts are the basic high-level forms for representing information in a knowledge base system. Each fact represents a piece of information that has been in the current list of facts. The knowledge base for the selection of MODM, MSKB, includes several groups of facts that have different functions. The basic knowledge about each MODM method and its various characteristics are described by a group of facts. Another group of facts is to relate the response of each question to the facts to be asserted by the inference engine into the working memory. It is also needed to get a number of facts to relate each characteristic to its corresponding question. The next group of facts relate to follow-up questions to follow given responses. It is necessary to get a set of facts to relate facts that are grouped under the same class. The last set of facts is used to initialise the inference process.

The knowledge is represented using 'def-templates' and 'def-facts'. Every 'def-facts' defines directly a fact. A def-template defines a group of related fields in a pattern similar to the way in which a record is a group of related data. Definitions of three pieces of def-templates and def-facts are shown in the following code:

- ***method:*** seven MODM methods and their various characteristics:

```
(deffacts  Method1
        (Method
          (Number 1)
          (Name ESGP)
          (Char1 interaction)
          (Char2 subset)
          (Char5 d-selection)
          (Char6 analyse)
          (Char7 ideal)
        )
)
```

- ***response:*** A set of facts relating the response of each question to the facts to be asserted by the inference engine into the working memory;

```
(deftemplate Response
        (field Question
          (type INTEGER)
          (default ?NONE)
        )
        (field Answer
          (type INTEGER)
          (default 0)
        )
)
```

- ***characteristics-question:*** A set of facts relating each characteristic to its corresponding question:

```
(deftemplate Char-to-Quest
        (field Char
          (type SYMBOL)
          (default ?NONE)
        )
        (field Quest_No
          (type INTEGER)
          (default ?NONE)
        )
)
```

6.5.3 Design of Rules

Rules are used to represent the heuristics to specify a set of actions to be performed for a given situation. This study defined a set of rules that collectively work together to solve the method selection problem. The method selection knowledge base system attempts to match all the characteristics of a method to those already asserted into the working memory. If the match failed, a characteristic which has the least weight will be discarded. A method will be selected if all its characteristics (or after missing) are found in the working memory. This study has also incorporated many heuristics that assist the system in the conflict resolution phase of the inference. For example, the rule to inform the user that a suitable method has been found shall have priority over other rules. Definitions of two rules are shown in the following code:

- ***call-question***: a rule relating to get the questions' number and its responses' number:

```
(defrule get-question
      (declare (salience 10))
      ?v1 <- (Question (Number ?num1))
      (test (neq ?num1 -1))
=>
      (retract ?v1)
      (bind ?response (quest ?num1))
      (assert (Response (Question ?num1) (Answer ?response)))
)
```

- ***question-action:*** after asking the decision makers a question and getting the response, the rule checks and compares the question number, answer number and facts between facts question_answer_action and response. If the numbers of question and answer match one of the facts question_answer_action, the numbers of question and fact are asserted.

```
(defrule quest_action
   (declare (salience 20))
   ?v1 <- (Q_A_Action (Question ?num1) (Answer ?ans1) (Facts $?facts))
   ?v2 <- (Response (Question ?num1) (Answer ?ans1))
   (test (neq ?facts no))
=>
```

```
    (retract ?v1 ?v2)
    (assert (Data (Question ?num1) (Facts ?facts)))
)
```

All patterns must be satisfied by facts in the fact-list for the rules to fire. A program will not start running unless there are rules whose left-hand side (LHS)'s are satisfied by facts. The inference engine sorts the activation according to their salience. This sorting process eliminates the conflict of deciding which rule should fire next.

6.5.4 CLIPS Encapsulation Package

This project uses an expert system tool called CLIPS to implement MSKB system. CLIPS is a complete environment for developing expert systems and it provides the basic elements of an expert system such as fact-list, knowledge-base which contains all the rules, the rule-base, and inference engine that controls the overall execution of rules. There are three ways to represent knowledge in CLIPS. One is by rules, which is primarily intended for heuristic knowledge based on experience. The second one is by defining functions, which is primarily intended for procedural knowledge. Object-oriented programming rules, also primarily intended for procedural knowledge, may pattern match on objects and facts. The three knowledge representation ways are all used in MSKB system. CLIPS offers seven different modes of conflict resolution: depth, breadth, LEX, MEA, complexity, and random. This study uses "depth strategy" in our application. In the depth strategy, new activation is placed on the agenda after activation with higher salience, but before activation with equal or lower salience.

CLIPS is designed to be embedded within other programs. When CLIPS is used as an embedded application, the system must provide a main program to call CLIPS. The main system and other subsystem of IMODSS have been developed in Delphi environment. CLIPS is written in C and so it cannot be called directly from Delphi. The system applied a package that contains the files for encapsulating CLIPS in Delphi through the use of DLLs. The system then created a new component "Tclips" in Delphi to encapsulate the Clips support files, which includes CLIPS.pas, CLIPSSupport, ClipsFact, CLIPSHeader and CLIPSrules. Through this encapsulated system support, various CLIPS' functions can be called in the Delphi based IMODSS main system.

CLIPS.pas is built to encapsulate the CLIPS DLL. Through creating procedures AssertFactHandler, RetractFactHandler, ClearHandler, PeriodicHandler, ResetHandler and so on, this program can be used to callback the function for asserting fact events, retract fact events, clear event, periodic events, and reset events. The following are two parts of this program.

```
(1) TFactEvent = procedure ( Sender: TObject; Fact: TClipsFact ) of Object;

 TClips = class( TComponent )
 { Only 1 Clips component per application can be used right now, otherwise, they
trash the ClipsObject variable. }

 public
   { Public declarations }
   function AssertString( FactString: String ): TClipsFact; virtual;
   procedure Initialize; virtual;
   procedure KillCLIPS; virtual;
   procedure Clear; virtual;
   procedure Reset; virtual;
   function Run: Integer; virtual;
   function Step: Integer;
   function StepBy( Value: Cardinal ): Integer; virtual;
   constructor Create( AOwner: TComponent ); override;
   destructor Destroy; override;
   procedure LoadBinaryFromFile( const FileName: string ); virtual;
   procedure LoadFromFile( const FileName: string ); virtual;
   procedure SaveBinaryToFile( const FileName: string ); virtual;
   procedure SaveToFile( const FileName: string ); virtual;
   property Facts[ Index: Integer ]: TClipsFact read GetFact;
   property FactCount: Integer read GetFactCount;

 published
   { Published declarations }
   property Filename: TFilename read FFilename write SetFilename;
   property Initialized: Boolean read FInitialized write SetInitialized;
   property OnAssertion: TFactEvent read FOnAssertion write FOnAssertion;
   property OnClear: TNotifyEvent read FOnClear write FOnClear;
   property OnPeriodic: TNotifyEvent read FOnPeriodic write FOnPeriodic;
   property OnRetraction: TFactEvent read FOnRetraction write FOnRetraction;
   property OnReset: TNotifyEvent read FOnReset write FOnReset;
```

```
  property Strategy: TClipsStrategy read GetStrategy write SetStrategy;
 end; { TClips }
```

```
(2) procedure AssertFactHandler( Fact: Pointer ); cdecl;    { Callback function for
assert fact event }
procedure RetractFactHandler( Fact: Pointer ); cdecl;   { Callback function for retract
fact event }
procedure ClearHandler; cdecl;                    { Callback function for clear event }
procedure PeriodicHandler; cdecl;                 {Callback function for periodic event}
procedure ResetHandler; cdecl;                    { Callback function for reset event }
```

In CLIPSFact, TCLIPSFact is defined as a Class in order to encapsulate the extraction of data from a CLIPS fact. A part of the coding of CLIPfact is shown as follows:

```
type

 TClipsFact = class(TObject)

 private
  { Private declarations }
  FAsserted: Boolean;    { is fact asserted yet?  TRUE except when we call CreateFact
and have not called assert yet }
  FFactPointer: PFact;    { Pointer to fact this class encapsulates }
  FOrdered: Boolean;      { Whether fact is ordered or not }
  FOrderedFields: TStrings;{ List of fields in an ordered fact, i.e., (mnem 1 2 3) }
  FSlots: TStrings;       { Slot names }
  function GetIndex: LongInt;
  function GetSlot( Index: String ): Variant;
  function GetSlotExists( Index: String ): Boolean;
  function GetText: String;
  procedure PutSlot( Index: String; Value: Variant );
  procedure SetAsserted( Value: Boolean );
 protected
  { Protected declarations }
  procedure SetFactPointer( Value: PFact ); virtual;
 public
  { Public declarations }
  function AssignSlotDefaults: Boolean;
  function Assert: Boolean; virtual;
  constructor Create( FactPtr: PFact ); virtual;
  constructor CreateFact( ADefTemplate: PDefTemplate ); virtual;
  destructor Destroy; override;
  function Retract: Boolean; virtual;
  property FactPointer: PFact read FFactPointer;
```

```
  property Slot[ Index: String ]: Variant read GetSlot write PutSlot;
  property SlotExists[ Index: String ]: Boolean read GetSlotExists;
published
  { Published declarations }
  property Asserted: Boolean read FAsserted write SetAsserted;
  property Index: LongInt read GetIndex;
  property Ordered: Boolean read FOrdered;
  property OrderedFields: TStrings read FOrderedFields;
  property Slots: TStrings read FSlots;
  property Text: String read GetText;
end; { TClipsFact }
```

6.5.5 Framework and Working Principle of Intelligent Guide Subsystem

All questions concerning method selection are shown on a DELPHI interface. After inputting these responses by users, including all answers for the questions and their degree of importance, the intelligent guide subsystem puts them in the response subsystem and ICS subsystem respectively. The data in the response subsystem will be then converted into the CLIPS-facts and is asserted to the fact base as the decision maker's response facts. The knowledge base attempts to match the characteristics of a method to the characteristics already asserted in the working memory. A suitable method is found once all its characteristics are matched with those in the working memory. The data in weight-array will be applied to find an *n-step* match method when a RC completed matched method doesn't exist.

The preliminary work is very important, however. Before running CLIPS and the inference engine, the intelligent subsystem first links with CLIPS. A support program TclipsFact is then embedded in this intelligent subsystem so that all CLIPS' functions and operations can be executed in a DELPHI environment. A DELPHI-CLIPS interface program is used to support the execution of the CLIPS operations in the DELPHI working environment and to provide a user interface. This study doesn't build any CLIPS' user interfaces because (1) the design ability provided by DELPHI for interfaces is better than CLIPS; and (2) a software package should keep a unified style of interface. In this DELPHI-CLIPS interface program, the intelligent subsystem can assert a set of facts by a public method or function, such as AssertString through the TClips code. The subsystem also can use FactCount and Fact properties for getting all the facts in the fact base, such as the Assert and Retract method to assert and retract a fact. The Tclips component also has a set of events to be used. They can be used to monitor CLIPS and its execution.

When CLIPS is called, the intelligent guide subsystem first checks if the CLIPS supporting files are in the correct location. The subsystem then calls "InitializeCLIPS" to initialise CLIPS. The subsystem again calls procedure "Clear" to clear the fact base. The next step is to load the CLIPS file that includes all fixed facts and rules. After this file is loaded, the subsystem executes "Reset procedure" and all fixed facts are entered into the agenda. The last function, "Run" is then called. All responses of the decision makers will be converted into the facts and the intelligent subsystem asserts them in the fact base. At the same time, the rules are fired and the subsystem starts an inference process. CLIPS attempts to match the patterns of rules against facts in the fact-list. If all the patterns of a rule match the facts, the rule is activated and put on the agenda. The agenda is a collection of activation that is those rules that match pattern entities. The intelligent guide subsystem and its principle of working are shown in Figure 6-5.

6.5.6 Method Selection Procedure under Intelligent Guide

Based on the working principles of the intelligent subsystem, Figure 6-6 shows the selection process of the most suitable MODM method in this IMOGDSS, which is self explanatory.

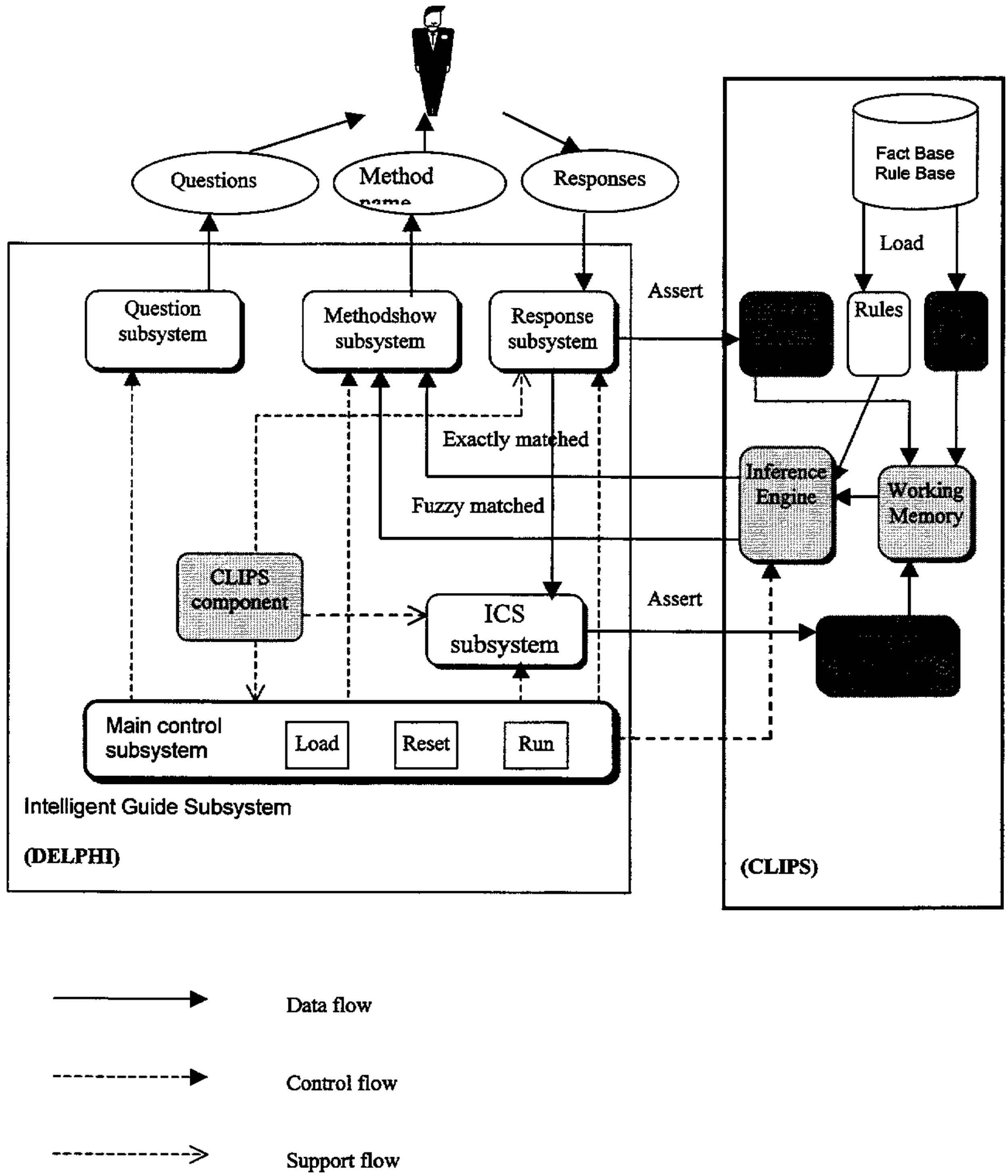


Figure 6-5 Intelligent guide subsystem and its working principle

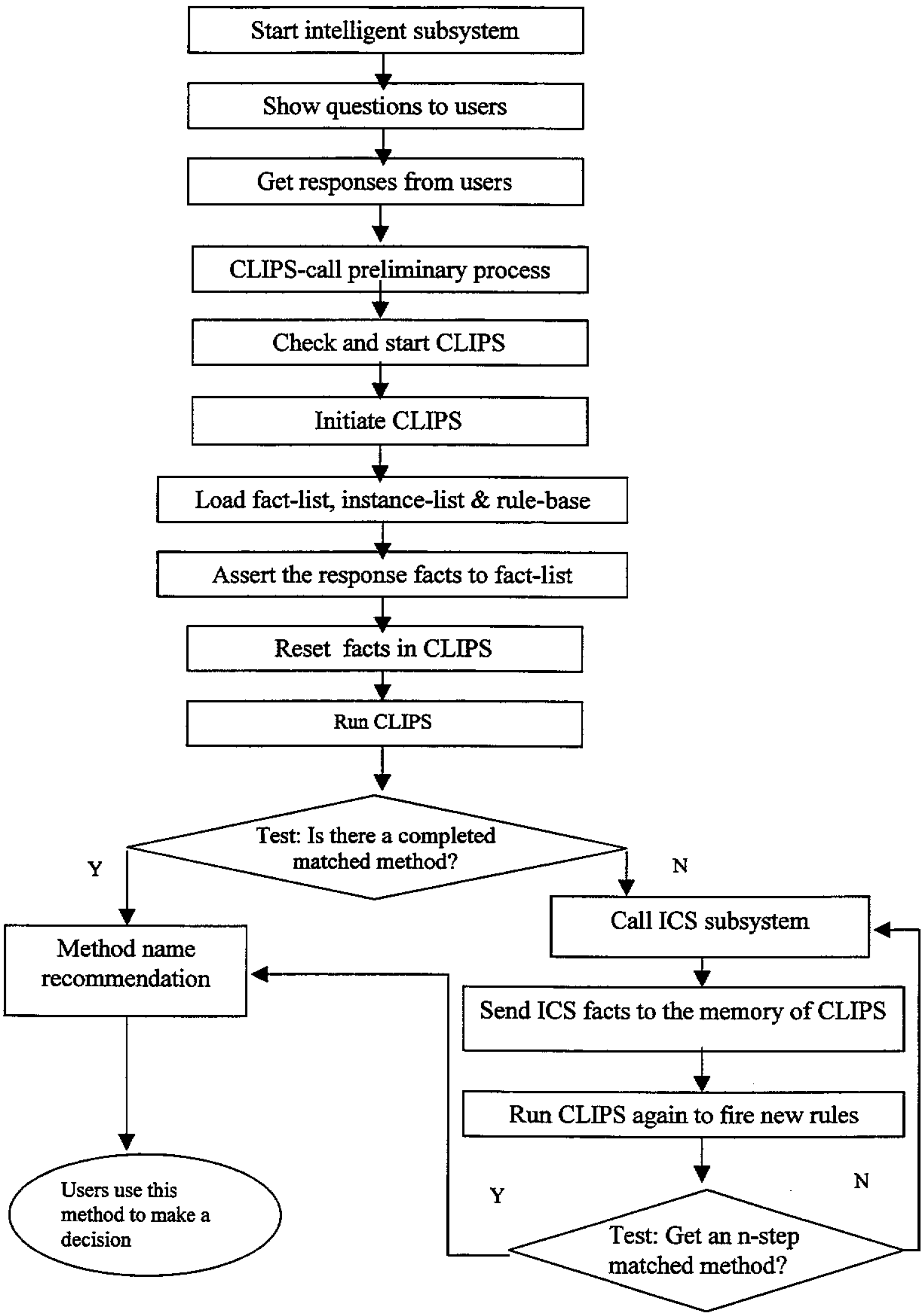


Figure 6-6 Method selection process

6.6 An Illustrative Example of Intelligent Guide Process

Consider an MODM problem that will be discussed in detail in Chapter 8. A decision maker X is assumed to be a marketing officer of this company and to be a novice at applying this IMOGDSS. The decision maker X needs to get a recommendation from the intelligent guide system for determining a suitable MODM method for his decision problem. X runs the intelligent guide subsystem and answers all questions. The part of his responses for the eight questions and weights for each question are shown in Figure 6-7 to Figure 6-10. After 'Finish' button is clicked, the system goes to the knowledge base supported by CLIPS. The system couldn't find any method to match completely X's needs. So the system applies 'Ignoring (missing) characteristic strategy' to find a 1-step matched method. Characteristics6 (analysis) has the lowest weight and so it is ignored. A 1-step matched method named STEUER is then found and recommended (Figure 6-11).

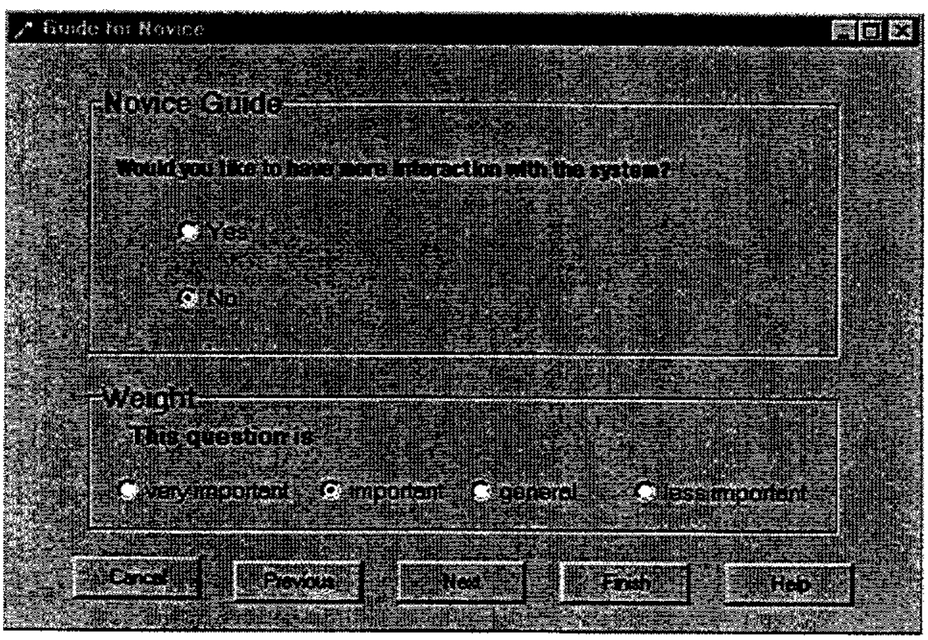


Figure 6-7 Question box 1 and decision maker X' response under the novice mode

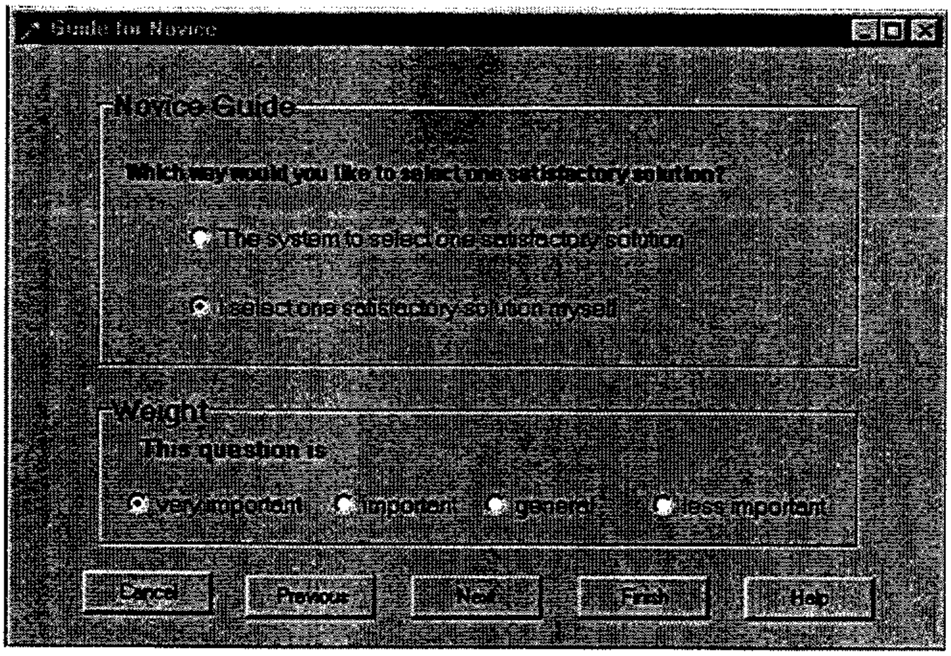


Figure 6-8 Question box 3 and decision maker X' response

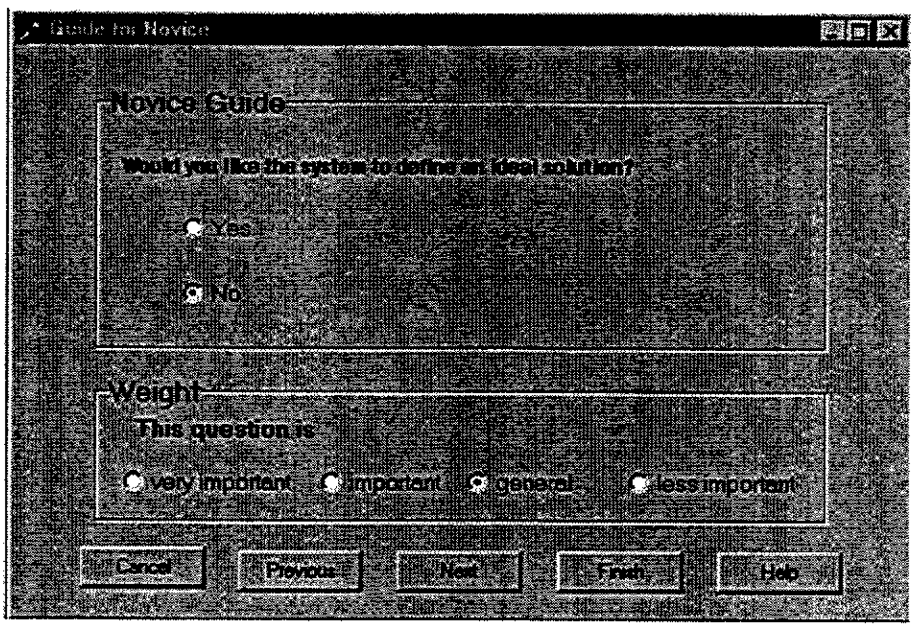


Figure 6-9 Question box 5 and decision maker X' response

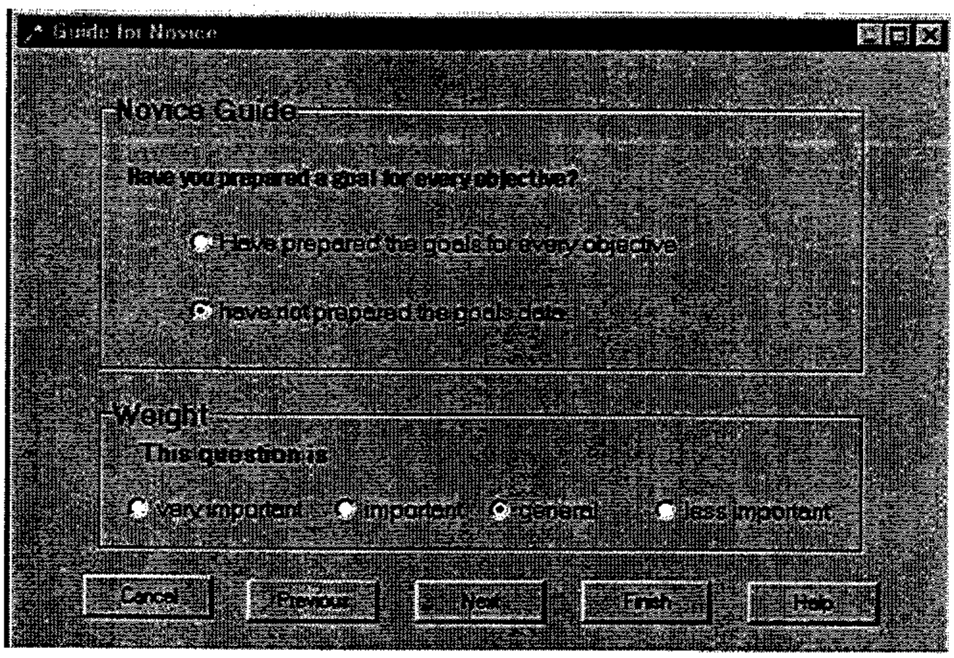


Figure 6-10 Question box 7 and decision maker X' response

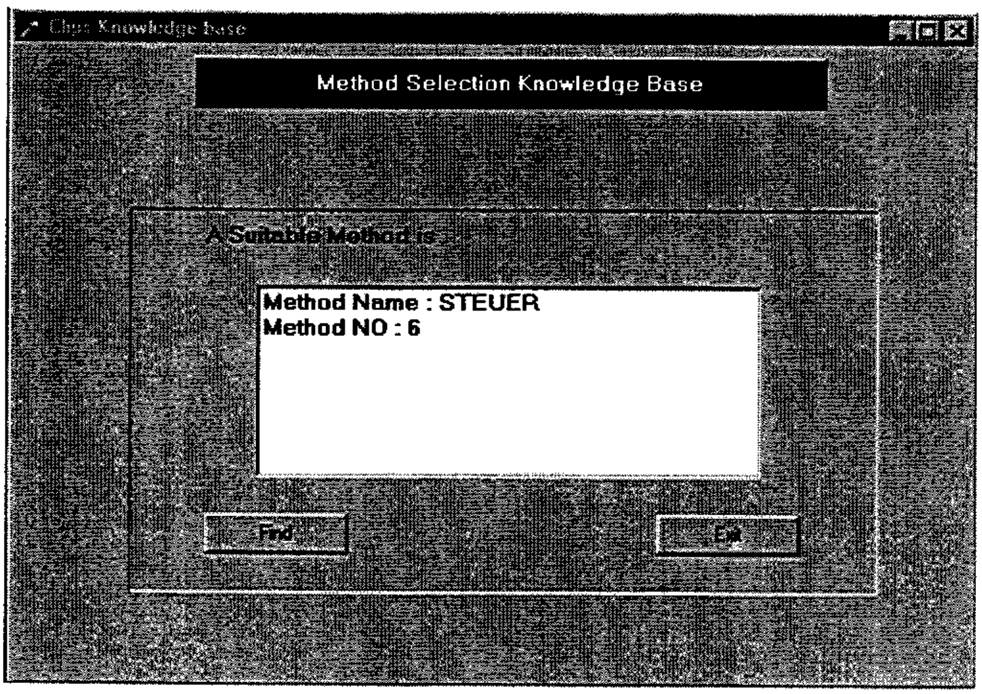


Figure 6-11 A 1-step match method is recommended

6.7 Summary

The major aim of integrating ES with MODSS and GDSS, in this study, is the provision of intelligent capabilities by adding an intelligent component to IMOGDSS. This chapter proposed the process of identifying, acquiring, and modelling the knowledge of MODM method selection, explored the possibility of embedding such a knowledge based intelligent guide within MOGDSS, discussed the match strategy and inferential process, and outlined a specific framework for developing such a knowledge-based intelligent guide.

The role of this intelligent guide subsystem is to provide a method selection function for decision making by complementing the decision makers' abilities and knowledge of MODM methods identification. This intelligent guide subsystem presents a decision maker with a series of questions and answers to choose from, thus determining which method is best suitable to the decision makers. By providing a decision maker with the ability of finding the best suitable method by getting the weights from the decision maker or by the system generating and executing ICMS, the intelligent subsystem has the flexibility to adapt to a decision maker's situation and style for the solution of an MODM problem.

Chapter 7

AN INTEGRATED MULTIPLE OBJECTIVE DECISION MAKING METHODOLOGY BASE AND ITS IMPLEMENTATION[5]

7.1 Introduction

Integrating multiple objective decision making (MODM) method management function is one of the major functional integration tasks in IMOGDSS. As a main component of IMOGDSS, MODM methodology base is used to provide MODM problem solution through various types of interactive processes. A methodology base is very important in a DSS because the type, features and algorithms of methods (some literature call models) in the methodology base determine directly the possible purpose, objectives and scope of applications provided in the DSS. Essentially, a methodology base is the core of a DSS and reflects the major characteristics of the DSS. The capability of the

[5] This chapter is based on papers published at the following conference proceedings:
(1) Lu, J., M.A. Quaddus and R. Williams," A framework and prototype for intelligent multiple objectives decision support system", *Proceedings of Asia Pacific Decision Sciences Institute Conference,* Shanghai, June 9-12, 1999.
(2) Lu, J., M.A. Quaddus and R. Williams, "Developing a knowledge-based multi-objective decision support system", *Proceedings of Hawaii International Conference on System Sciences (HICSS),* 4-7 January, 2000.

methodology base including its effectiveness and flexibility will determine the degree of functionality of a DSS.

This chapter proposes the architecture and features of MODM methodology base, displays its data structure, interaction types and central role in IMOGDSS for supporting decision making. This chapter suggests a DSS approach to implement effective MODM with a methodology base. This chapter also presents a GUI based design and implementation of the MODM method subsystem which allows selective and flexible use of seven popular MODM methods to solve various linear multiple objective decision problems. The chapter then describes the active uses of interactive techniques of the seven MODM methods and various GUI to improve the performance and effectiveness MODM. The seven MODM methodologies, namely, ESGP, ISGP, IMOLP, LGP, STEM, STEUER and ZW are addressed respectively in each section of this chapter for their working principle, characteristics, algorithm and application procedure. With professional programming and incorporation of GUI, this subsystem will allow more users to apply MODM methods to their decision problems without being overwhelmed by the underlying mathematics. This subsystem also becomes a learning process for the decision makers where the decision makers can gain new insights about their decision problems and then understand the exploration of the solution and thus contribute in developing the application model.

7.2 Architecture and Features of MODM Methodology Base

7.2.1 MODM Methodology Base and Method Subsystem

This method subsystem aims to manage an MODM methodology base that provides a set of MODM methods selected to support the decision makers who have different requirements for the solution procedure in order to solve different multiple objective decision problems. The decision makers can try any method in the methodology base and freely change their preferences in order to evaluate different scenarios in an interactive manner. Currently seven methods are available, namely, ESGP, ISGP, IMOLP, LGP, STEM, STEUER and ZW. However, the flexibility of the subsystem's modular structure will allow any new MODM method to be easily integrated into this methodology base.

These seven methods are developed into independent executables to facilitate the flexibility required of the system. These methods share similar data acquisition routines and these routines are developed as independent modules so that data acquired could be accessed by all the methods. This also contributes to the flexibility of the system (Poh 1998). Figure 7-1 shows the structure of this method subsystem.

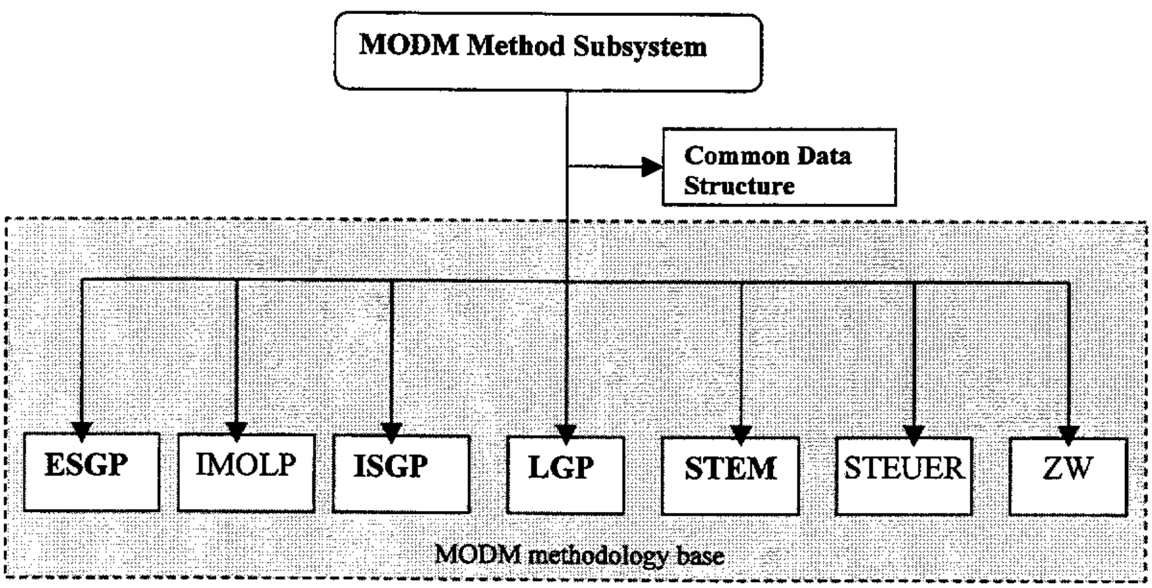


Figure 7-1 Architecture of MODM methodology base

7.2.2 Data Structure in MODM Methodology Base

The common data structure routines strictly keep some common data structures. Two matrices, called '*objective matrix*' and '*constraint matrix*', must be generated by *objective grid* and *constraint grid*, and are described in the common data structure routines. The objective matrix has the rows denoting the objectives and the columns the decision variables of the MODM. The constraint matrix has the rows representing the constraints and the columns will represent the decision variables as before. In a similar way to the two matrices, the results of an MODM application are also saved as a common data structure. This methodology base applies a *File Stream* approach to manage all these common data structures. The following is the main common data structures and common variables used in this methodology base.

```
type DecVariableInput1=record
   NoVariables: Integer; {number of variables}
   NoObjectives: Integer; {number of objectives}
   NoConstraints: Integer; {number of constraints}
   VariableNameArray: array[1..Maxvar] of string[8]; {variable names}
   ObjectNameArray: array[1..MaxObj] of string[8]; { objective function names}
   end;

type ObjectiveMatrixinput1=record
   MaxMin: array[1..MaxObj] of -1..1; {Max,Min, OT, -1..1}
   ObjectiveVarData: array[1..Maxvar, 1..MaxObj] of real; {objective functions}
   end;

type ConstraintMatrixInput1=record
   ConstraintVarData: array[1..Maxvar, 1..MaxCon ] of real; {constraint functions}
   SignTypeArray: array[1..MaxCon] of signtype; {<,=,>, >=, <=, TS}
   ConstraintValue: array[1..MaxCon] of real; {Right Hand Side}
   end;
```

```
type result1=record
   outputF: array[1..MaxObj] of real; { objective function values}
   outputX: array[1..MaxVar] of real; {variable values}
   end;

type Alldata =record
   DecVariableInput: DecVariableInput1;
   ObjectiveMatrixInput: ObjectiveMatrixInput1;
   ConstraintMatrixInput: ConstraintMatrixInput1;
   Result: Result1;
   end;

var DataFile: TFileStream;
var TheData: Alldata;
    FileName: String;
    DataFile: file of TfileRecord;
    Filerecord: TfileRecord;
    end.
```

7.2.3 Features of MODM Methodology Base

This MODM methodology base is applied to produce a solution through an interactive and consultative process between the problem owners and the system, and sensitivity analyses play a major role in fine tuning the users' requirement for the objectives. In the interactive process, some decision makers prefer to define clearly and input their goals and other criteria explicitly. Very often, decision makers do not wish to make their criteria or preferences explicit. They do not separate value and facts but, by judgment, they make a choice. Therefore, some methods included in this method base are more suitable for the first type of requirement and some are efficient for the second type of requirement. This is one of the advantages of the methodology base containing a sufficient number of methods for the decision makers.

The method subsystem is activated through invoking the *method* menu shown in Figure 5-3. The menu produces a list of available methods from which any method (currently seven of them) can be selected. The subsystem then shows a short description of the selected method, which we call the '*home page*'. A button oriented interactive solution procedure is then started. At any point in any method, the decision maker is allowed to go back to the main menu if he/she is not happy with the currently selected method.

7.2.4 User-Driven Interactive Process Framework

A solution is not generally made in a single step but is in an interactive trade-off process, where the number of steps depends on each method's algorithm and users' interest in exploring a satisfactory solution. However, this subsystem presents these MODM methods under a similar framework for data collection, decision solution analysis and the satisfactory solution obtained, in order to facilitate users to solve their decision problems through multiple MODM methods. The framework involves the following functions that correspond to one or more buttons driven by user:

- Goal Enter---entering a set of goals, weights and priorities of objectives;
- Solve---checking constraints and identifying efficient solution;
- Solution Explore---solution analysis, involving a relaxation process for increasing the values of some objectives and decreasing others, a set of ideal solution is usual shown;
- Solution Choose---a series of alternative solutions' show and choice selection;
- Final Solution---a satisfactory solution is determined; and
- Finish---saving this satisfactory solution into a database and ending this solution procedure.

Each method takes several or all of these functions to complete a solution procedure. These functions are designed as button-driven procedures and used to primarily manage the interaction process. There are three common functional buttons *'Solve'*, *'Final solution'* and *'Finish'* under each method. Based on Bui's (1984) Input-Tool-Output Relation Model in MCDM, Figure 7-2 shows a DSS approach to implement effective MODM with a methodology base. It describes the active uses of various interactive functions to improve the performance of MODM in a DSS.

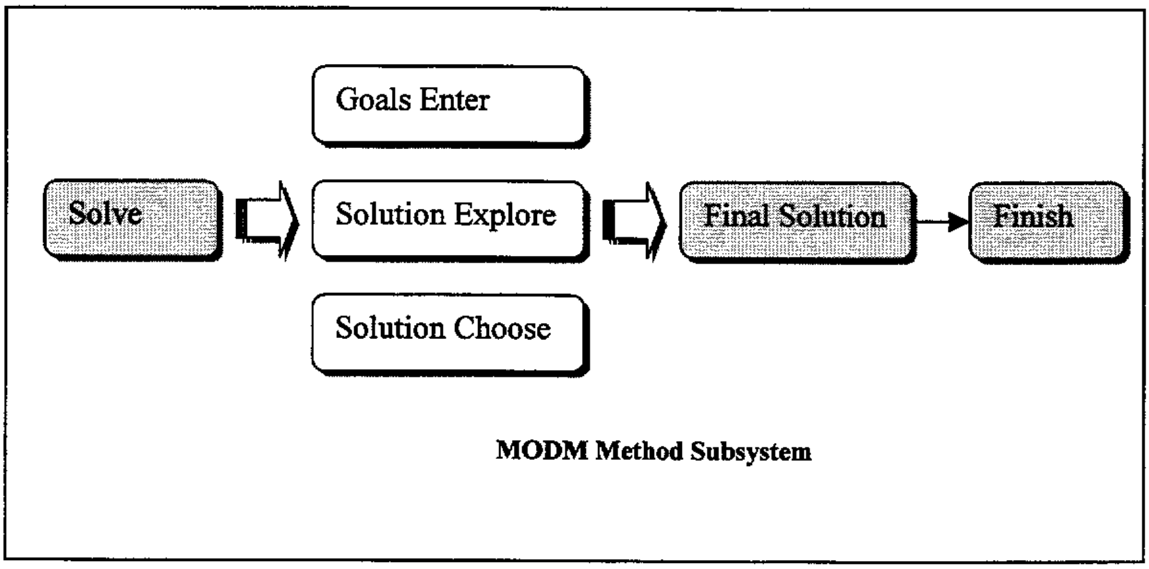


Figure 7-2 An interactive paradigm based DSS approach in MODM

Based on the approach described in Figure 7-2, the method subsystem is effective in allowing rapid progress in the production of efficient solutions and a solution analysis. Interaction is used to support the choice phase of the decision making process and thus can enhance user cooperation and reduced users' training requirements.

7.2.5 MODM as a Learning Process

It is widely claimed that the using an MCDSS can become a useful learning process for the decision makers (Steuer 1986; Shin & Ravindran 1991), who can then gain new insights about the problem and thus contribute in developing the application model. By implementing the interactive paradigm, this method subsystem provides such a learning process about the MODM solution, whereby the decision maker can learn to explore more solutions, analyse these solutions, recognise a satisfactory solution and understand the relative importance of factors in the application. It should be emphasized that this subsystem also involves characterising different kinds of decision situations and exploring the implications of these characteristics; describing different types of decision models available for analysing various decision situations; describing the processes of constructing decision models. As these concepts are related to the application of expert systems, they have been discussed in Chapter 6.

7.2.6 Implementation Strategy of MODM Methodology Base

The seven MODM methods in the methodology base all address the multiple objective linear programs (MOLP). MOLP is the most commonly studied problem in MCDM (Korhonen, Moskowitz & Wallenius 1992). Dozens of methods have been developed for solving MOLP problems. In early methods, a common feature was to operate with criterion weights, limiting consideration to efficient extreme points. Today, many systems are based on the use of aspiration level projections (Korhonen, Moskowitz & Wallenius 1992). These functions can be controlled either by varying weights (keeping aspiration) or by varying the aspiration levels (keeping weights fixed). The two features will be seen from the seven methods described in the following sections.

For constructing the methodology base, several implementation strategies are conducted in four aspects, as shown in Table 7-1.

Table 7-1 Implementation strategies

Implementation strategy	Purpose
Divide project into manageable pieces	To minimise the risk of producing a massive system that doesn't work
Keep the solution simple	To encourage use and to avoid scaring away users.
Hide complexity	The system is presented as a 'black box' that answers questions using procedures not presented to the user.
Meet user needs and institutionalise system	A system is to have many individual users in an ongoing application.

There are two specific cases that can be involved: (1). No solution exists in a decision problem. This situation needs to be identified by constraints checking at the beginning of use of a method or during the solution process using a method. (2). Decision makers may want to return to use another method during the working process of a current method. This MODM methodology base must handle these two cases.

In the following sections, brief overviews of the seven MODM methods are provided. Details can be found in the cited references.

7.3 Efficient Solution via Goal Programming (ESGP) Method

The ESGP method was proposed by Ignizio (1981). It is one of the most popular MODM methods. It finds an arbitrary efficient solution from the feasible solution space using goal programming technique. Interaction with the decision makers then begins. If the initial solution is not satisfactory, the decision maker is asked to give the amounts for each of the objective functions that he/she can relax in order to gain in at least one other objective. The feasible solution is then modified to include this information and the method then finds another efficient solution from the reduced feasible space. Each solution is offered to the decision maker for his/her response. The method either terminates with a satisfactory solution, or with no satisfactory solution found.

Advantages of goal programming are that the decision maker does not need to give the numerical weights for the objectives; he/she needs to give only an ordinal ranking of them. However, this implies a value trade-off assumption: that is, in minimising the function $h_{i+1}(d^+,d^-)$, the value preference for h_i is given over h_{i+1}. The function is also very sensitive to the goal vector set for the objectives and ordinal ranking given by the decision maker. It is noted that d^+,d^- are respectively the under-achievement and over-achievement, and $h_i(d^+,d^-), i=1,2,\cdots,l,$ is a linear function of the deviational variables and is called achievement function. Goal programming method has been widely used in many MODM methodologies such as ESGP, ISGP and LGP as a basic algorithm.

ESGP posses the following characteristics:

- Provides an ideal solution
- Uses a degradation process
- Displays all efficient points
- Allows users to select a satisfactory solution from a solution set

- Allows users to make multiple pro-interaction and post-interaction processes

This algorithm is implemented by using Delphi in the IMOGDSS. A flow chart of this method is presented in Figure 7-3.

In IMOGDSS, once the ESGP menu is chosen, the "home page" of ESGP is shown first (Figure 7-4 (1)). A decision maker can follow the following steps to solve his/her decision making problem:

Step 1. Press ***Solve*** button, the system examines the consistency of the constraints and forms the first solution.

Step 2. Press ***Solution*** button, the first solution and the ideal value of each objective is displayed. Users are asked to accept or reject the solution (Figure 7-4 (2)).

Step 3. If ***Accept*** button is clicked, then this solution is the final satisfactory solution and the execution of ESGP stops. However the decision makers have different preferences and different favourite choices, and they often like to further search so that more solutions are obtained. The ***Reject*** button is thus clicked and a relaxation process is invoked.

Step 4. After the ***Reject*** button is clicked, a message is shown "You have to accept the optimal solution found through goal programming. Do you want to find a set in the space adjacent to the current solution?" If the users agree to this, they can start to indicate the degradation of objective functions under the guidance of the system.

Step 5. The users will have to compromise the results among the objectives. An improvement in one or more of the objectives will result in the sacrifice of other objectives. Use +ve value for degradation and use -ve value for improvement. The screen displays a solution, ideal value and asks users to enter degradations. After a set of degradation values are entered, ***Continue*** is pressed (Figure 7-4(3)).

Step 6. If a degradation value exceeds the ideal value, an " exceed value error" message is shown. If all degradation values are negative, then a message " you must have at least one +ve degradation" is shown. Otherwise, a new group of solutions are formed. The first solution from the group is then shown (Figure 7-4(4)). The users can accept it or reject it.

Step 7. If the ***Accept*** button is clicked, then this solution becomes the final satisfactory solution and the ESGP stops. If the ***Reject*** button is clicked, another solution in this set of group is shown and a message is shown "Are you satisfied with this solution?". If ***Yes*** is chosen, this solution is taken as the final satisfactory solution otherwise another new solution is shown until a message "all efficient points satisfying your degradation limits have been presented" is shown.

Step 8. After the ***Accept*** button is pressed, a final solution is shown for confirmation. Once the ***Final solution*** button is pressed, the result will be automatically transformed into a model base as a current model. Users are then asked if they wish to save the solution.

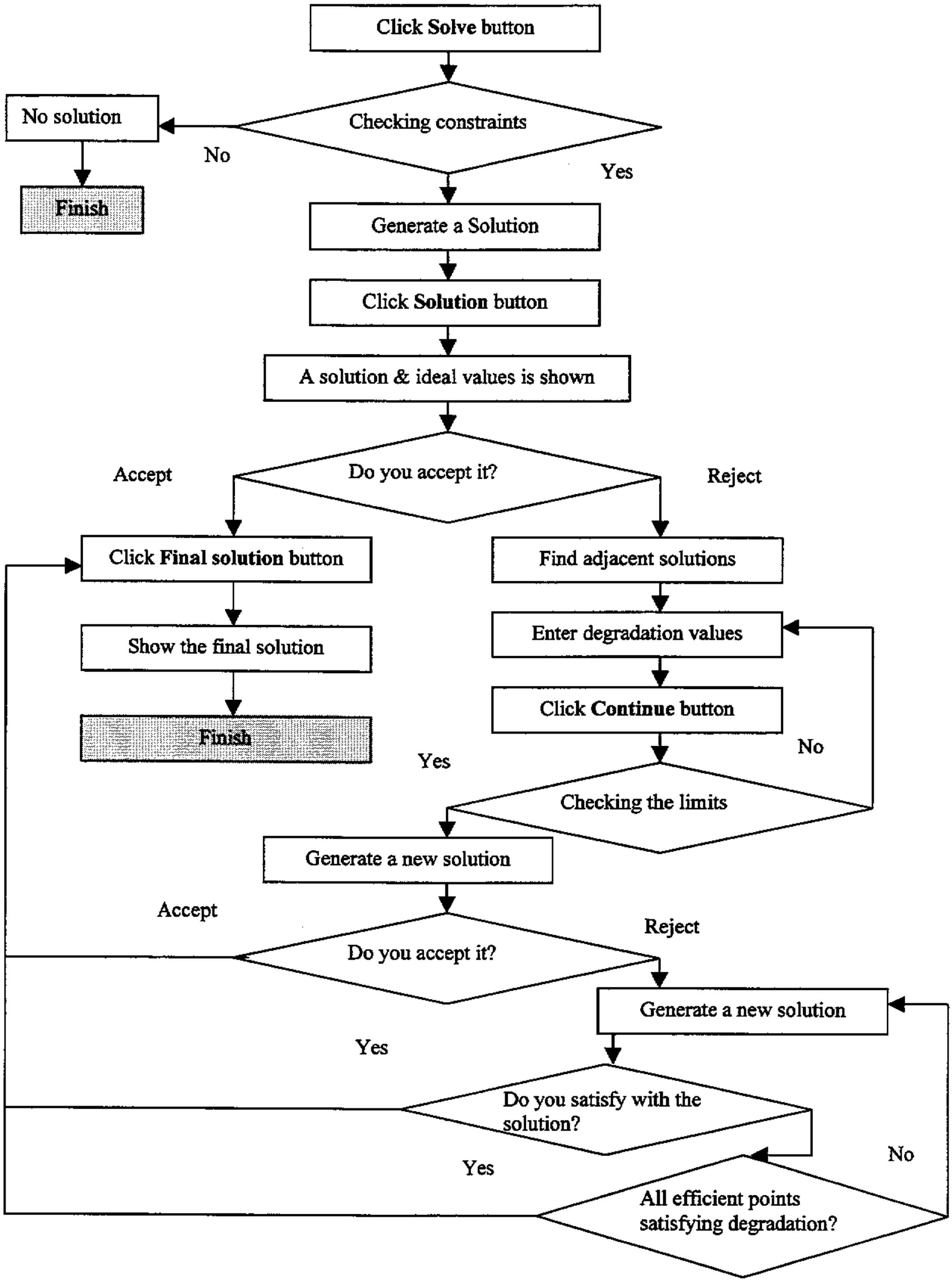


Figure 7-3 Algorithm of ESGP

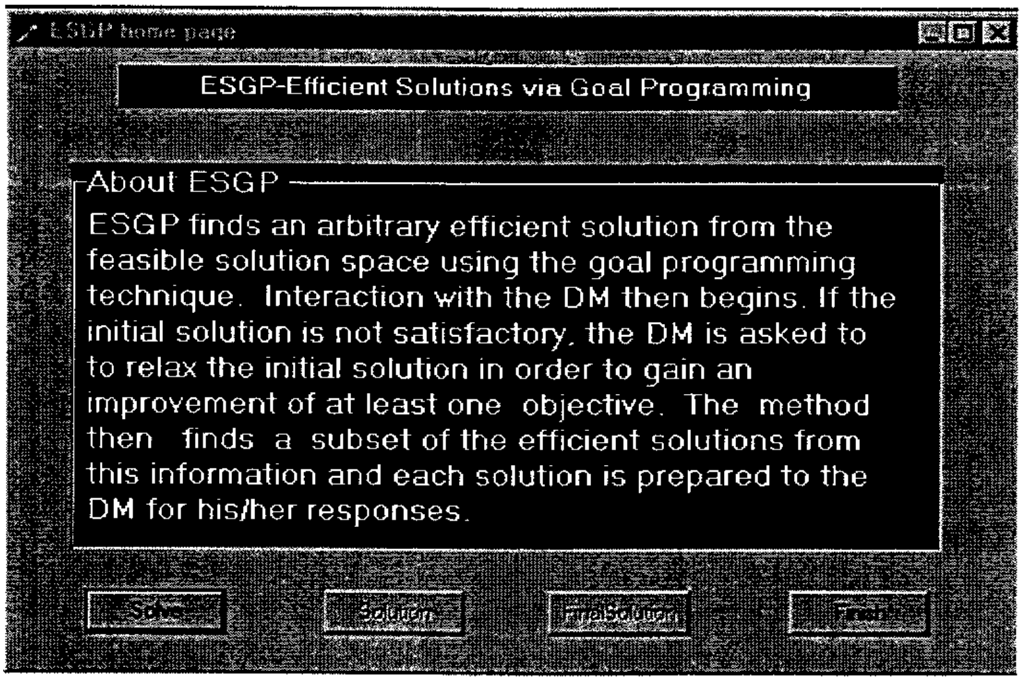


Figure 7-4(1) Home page of ESGP

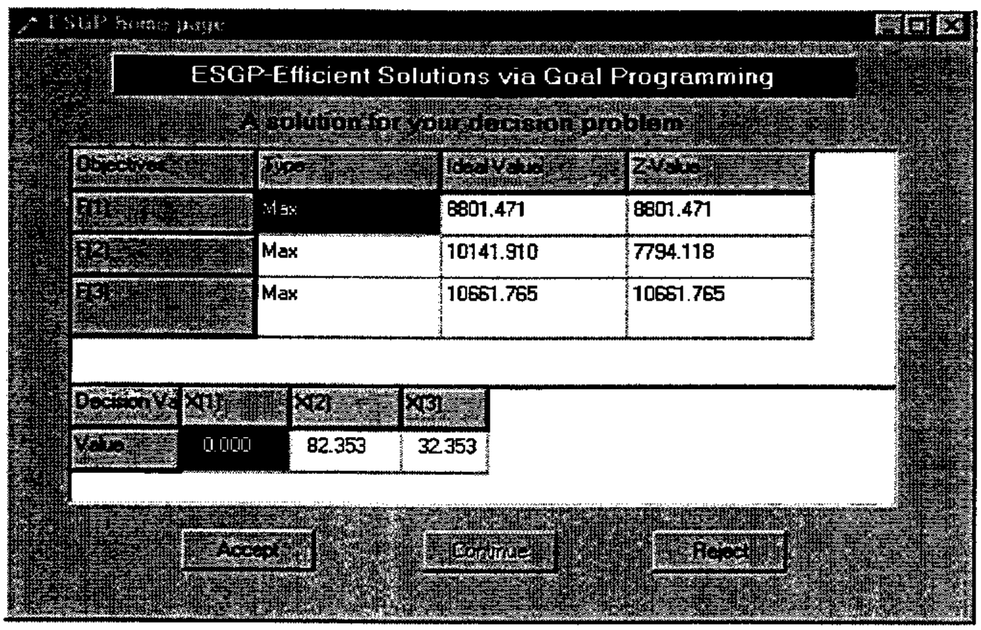


Figure 7-4(2) The first solution and the ideal value of each objective is displayed

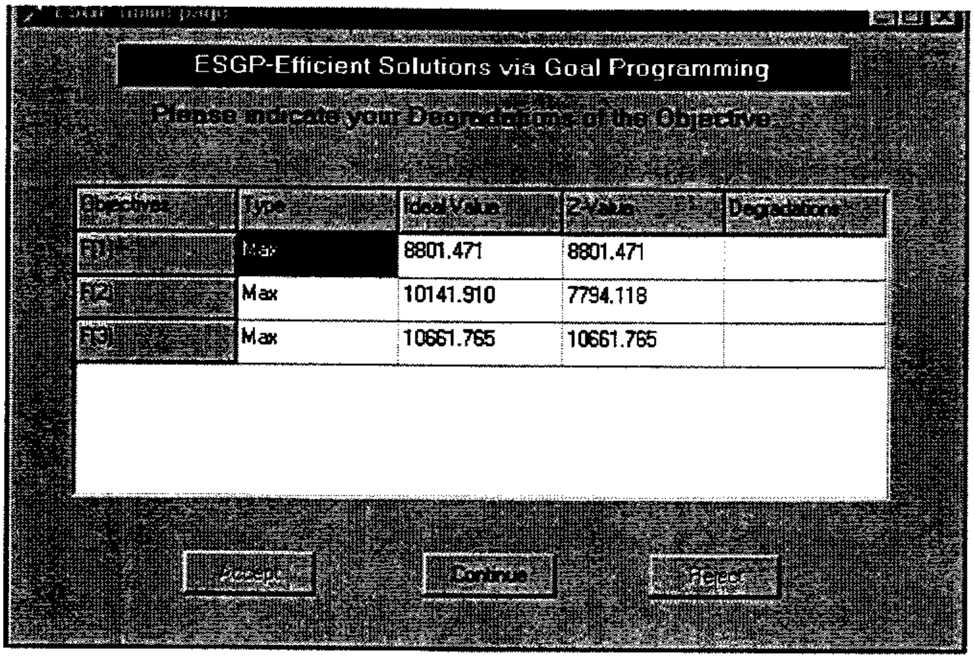


Figure 7-4(3) Degradation values are entered

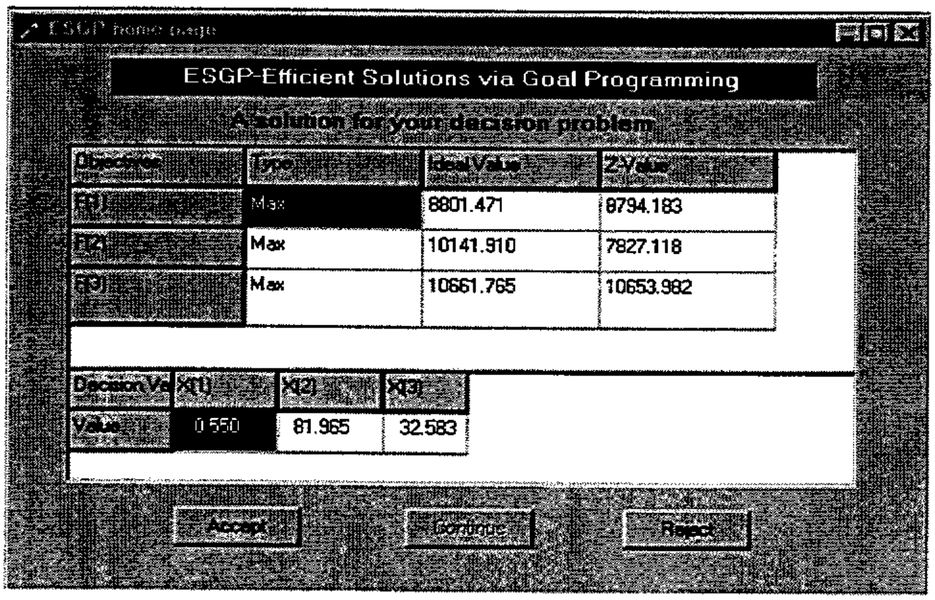


Figure 7-4(4) A new solution is formed

7.4 Interactive Multiple Objective Linear Program (IMOLP) Method

IMOLP (Quaddus & Holzman 1986) is an interactive method that explores the efficient solution (extreme point) of MOLP in a systematic way. The method requires little information from the decision maker and explores the efficient solution from different regions of the efficient surface. In MOLP the decision maker is required to provide local preference information in terms of either gain or sacrifice in different objectives without the need to specify any numeric value. This method then finds a set of efficient solutions that match the information from the set of adjacent solutions. The method stops if the satisfactory solution is found, otherwise the process is repeated. The flow chart of this algorithm is described in Figure 7-5.

IMOLP possesses very rich features of MOLP and has more solution exploring features:

- Users are required to enter weights for each objectives
- Provides multiple cycles to users to try different weights sets
- Uses a trade-off process to support users' solution analysis
- Provides more solution exploring and selecting means, such as a set of solutions or a unique solution, to users to reach a satisfactory solution
- Allows users to make multiple pre-interaction, pro-interaction and post-interaction

The performance process implemented in IMOGDSS is described as follows:

Step 1. After a "home page" of IMOLP is shown, users can click ***Weight*** button to enter the weights for each objective. A table that includes the name and the type of objectives is displayed to obtain the weights (Figure 7-6 (1)).

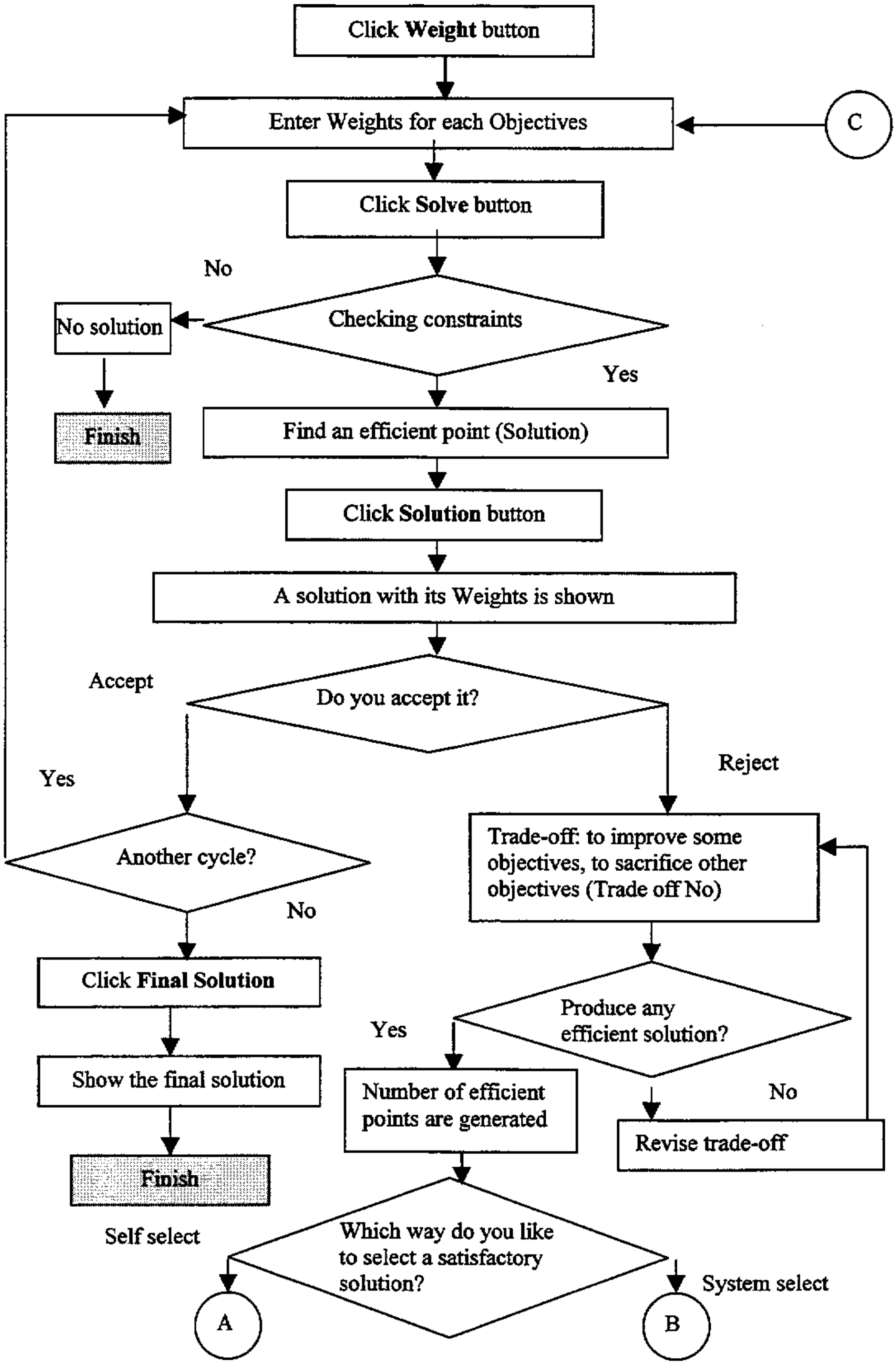
Step 2. Click ***Solve*** button, the system checks the constraints and finds an efficient point.

Step 3. After the ***Solution*** button is pressed, a solution is shown. Users are asked to select this solution as a satisfactory solution by pressing the ***Accept*** button or to reject this solution by pressing the ***Reject*** button (Figure 7-6(2)). If the decision maker presses the ***Accept*** button, the method asks the users if he/she wishes to consider another cycle. If the users want to explore another cycle, repeat Step 1 else a final solution is formed and this method stops. If the ***Reject*** button is chosen a trade-off process starts.

Step 4. The trade-off process is to improve the values of some objectives and to sacrifice other objectives. Here, ***G*** means objectives are improved and ***S*** means objectives are sacrificed. After users enter the trade-off values, the method checks if new efficient points are created. If no new efficient point is created the users need to revise the trade-off (Figure 7-6(3)). Once a trade off produces a set of efficient solutions (Figure 7-6(4)), users have two ways to select a satisfactory solution.

Step 5. The first way indicates that users can select a satisfactory solution by themselves. The system shows the set of solutions in turn. Users can choose one of them as a satisfactory solution. The ***Final solution*** button is then pressed and this method stops if she/he doesn't want to explore another cycle.

Step 6. The second way provided by this method is to produce the best solution by the method directly. If users are not satisfied with this solution, they are allowed to go to another cycle. Step 1 is then restarted.


Click Weight button
Enter Weights for each Objectives
C
Click Solve button
No
Checking constraints
No solution
Yes
Finish
Find an efficient point (Solution)
Click Solution button
A solution with its Weights is shown
Accept
Do you accept it?
Yes
Reject
Another cycle?
Trade-off: to improve some objectives, to sacrifice other objectives (Trade off No)
No
Click Final Solution
Produce any efficient solution?
Yes
Show the final solution
Number of efficient points are generated
No
Revise trade-off
Finish
Self select
Which way do you like to select a satisfactory solution?
System select
A
B

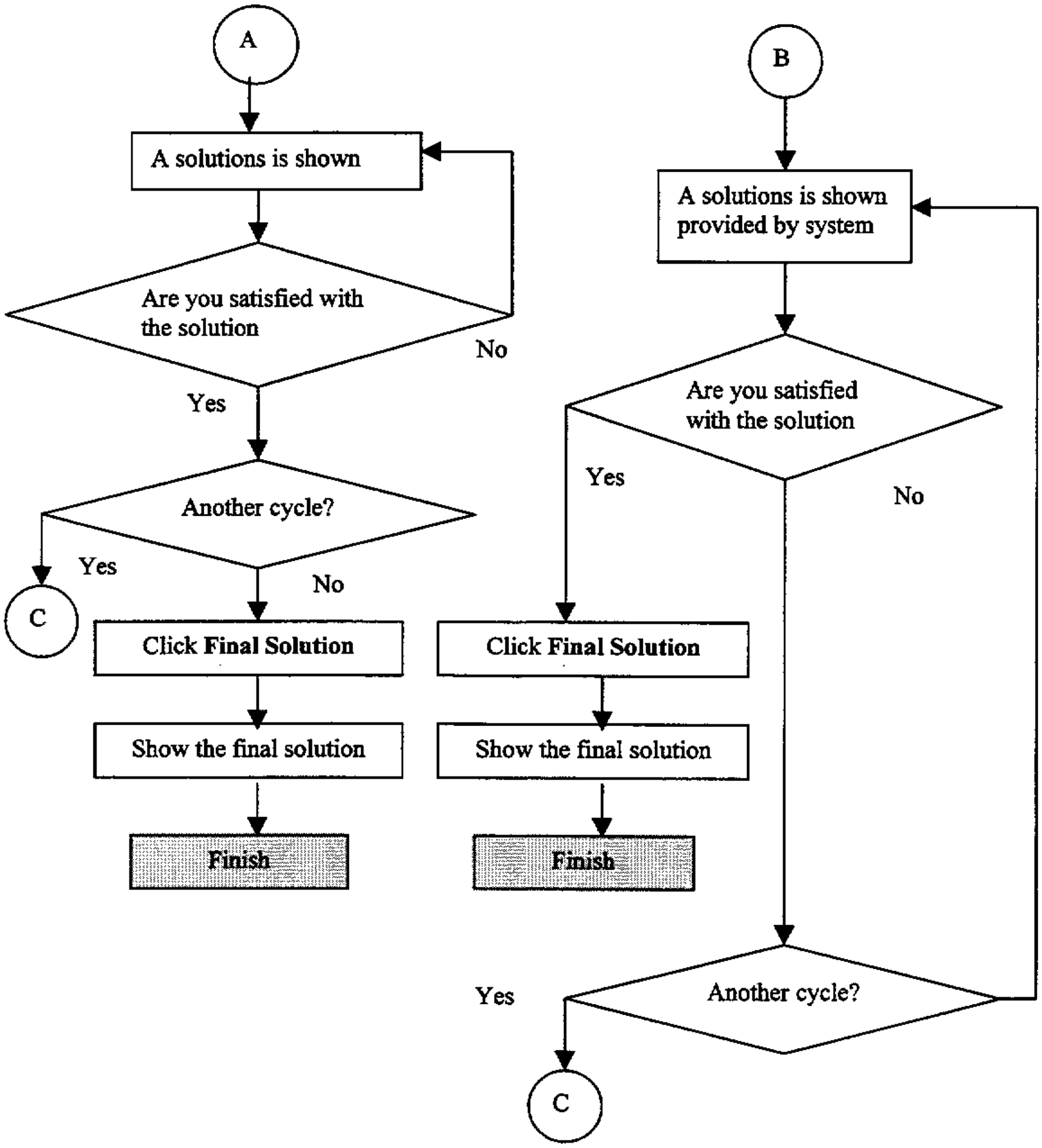


Figure 7-5 Flow chart of IMOLP algorithm

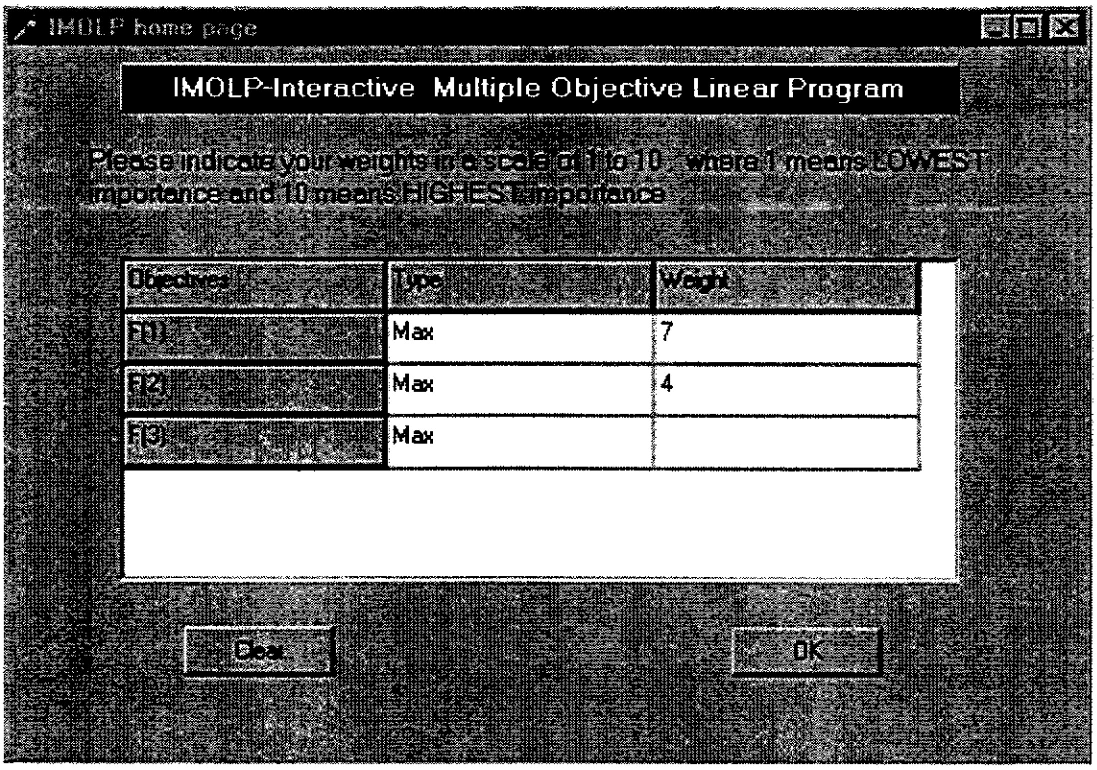


Figure 7-6(1) Entering the weights of objectives

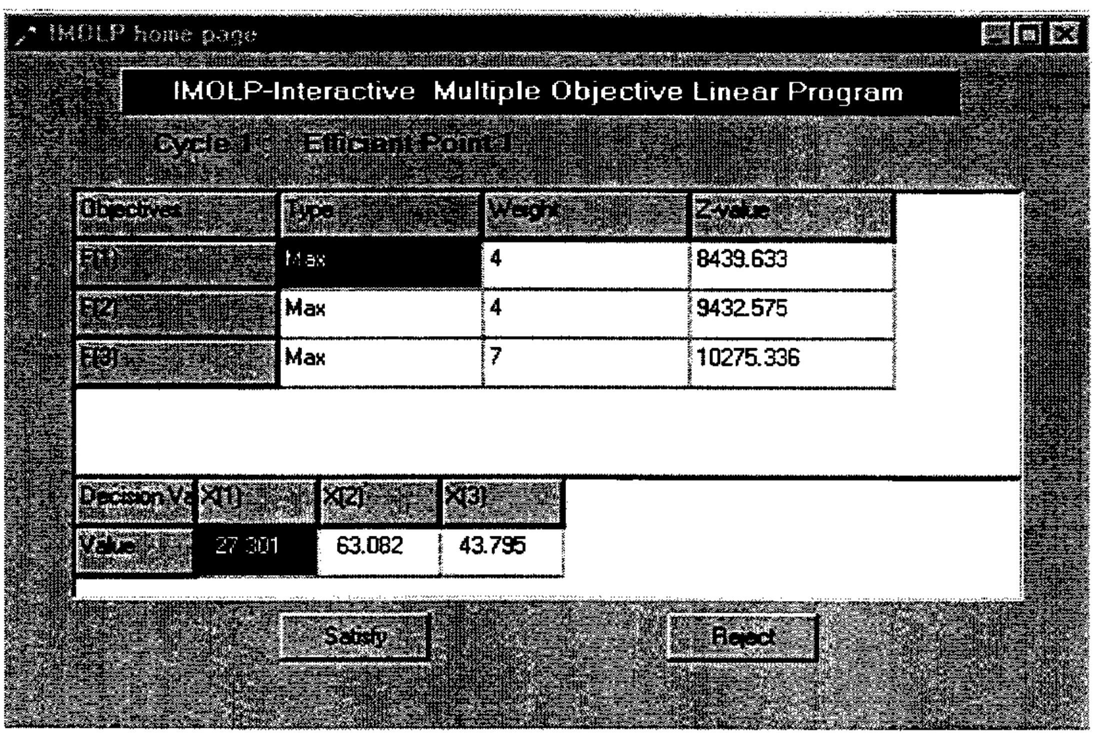


Figure 7-6(2) Select this solution or reject it

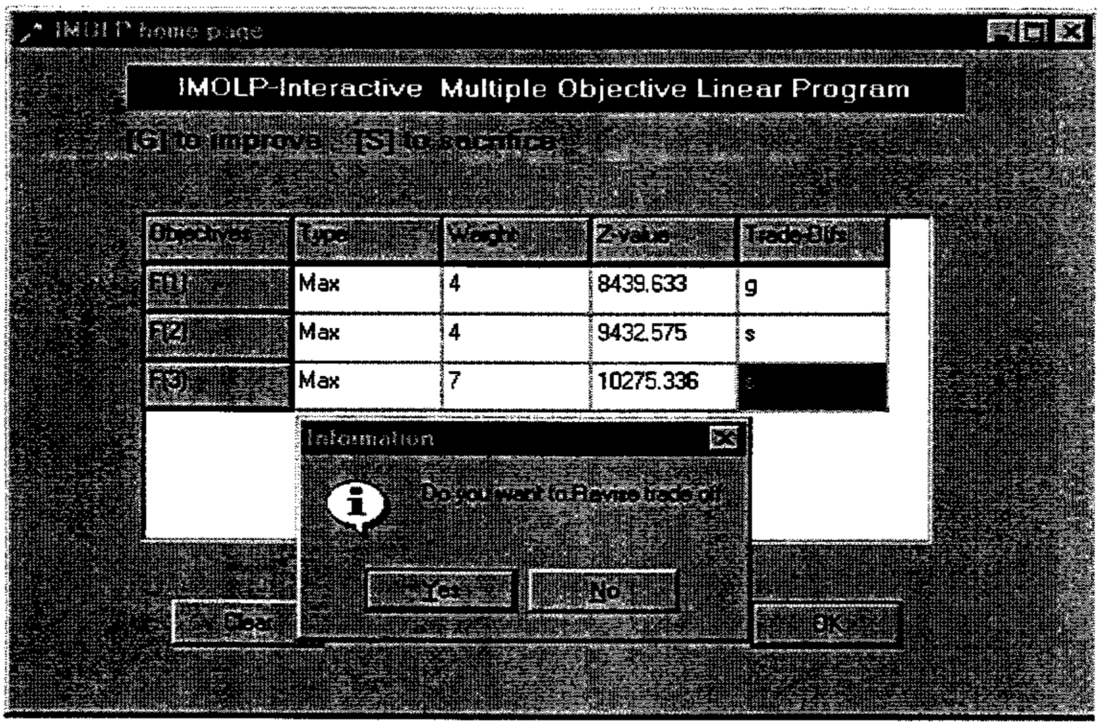


Figure 7-6(3) Revise the trade-off

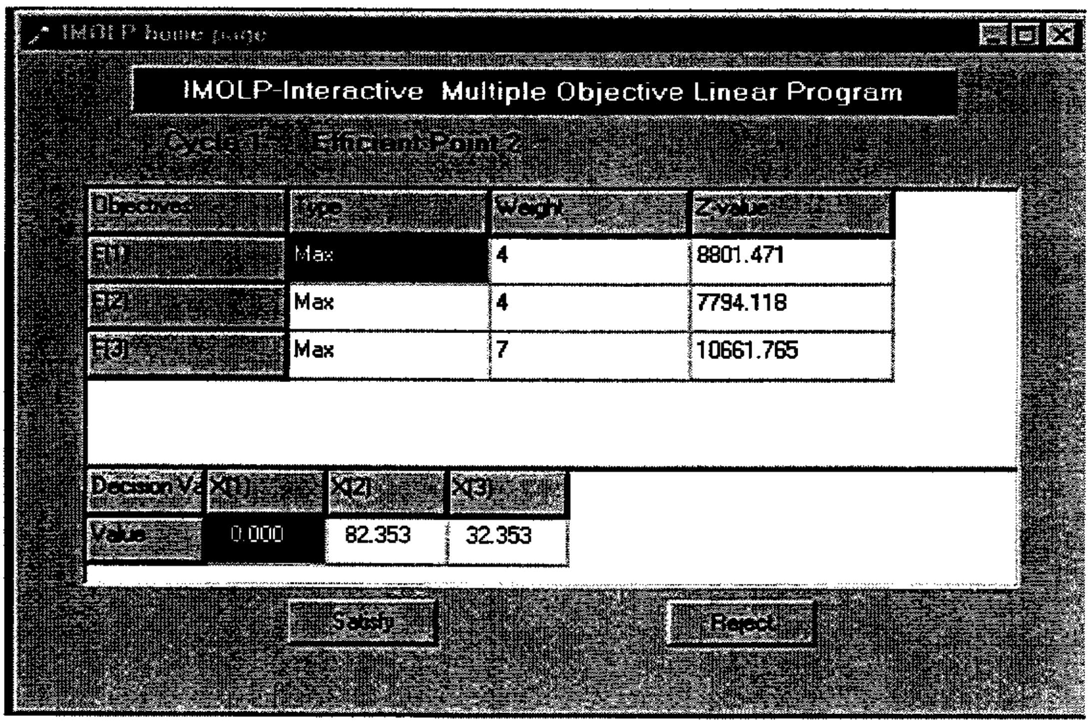


Figure 7-6(4) A new efficient point

7.5 Interactive Sequential Goal Programming (ISGP) Method

The interactive sequential goal programming (ISGP) method was presented by Masud & Hwang (1981). This method links goal programming with an interaction scheme that provides a subset of efficient solutions at a time. The method first presents ideal solutions and then interacts with the decision maker for her/his desirable goal levels for different objectives. This method then generates a subset of efficient solutions that satisfy the individual goals serially. If the decision maker chooses a satisfactory solution from the subset, the method stops, otherwise, it interacts with the decision maker again for his/her modified desirable goals and the method continues. It is found that this method can produce enough alternative solutions for interaction with a decision maker. The main advantage of this methodology is that a decision maker can easily see the achievable objectives i.e. how far the *return* and *risk* can be optimised. It quickly becomes a learning process for the decision maker. Figure 7-7 shows the algorithm of the method.

ISGP method is an extension of goal programming model with interactive strategies to solve MODM problems. It

- provides an ideal solution and worst solution
- require users to enter goal for each objective
- generates a set of solutions through explicit trade-off
- allows users to select a satisfactory solution from the solution set
- allows users to make multiple pre-interaction, pro-interaction and post-interaction

When ISGP is chosen in the main interface of IMOGDSS, the "home page" of ISGP is shown (Figure 7-8(1)). The decision makers can follow these steps to arrive a satisfactory solution:

Step 1. Click ***Solve*** button, the system starts to compute the ranges of each objective.

Step 2. The system then asks users to enter the goal for each objective. In order to avoid the infeasible goals input, a table that includes the name and the type of each objective, the ideal values and the worse values of these objectives is displayed (Figure 7-8(2)). The entered goals must be between the ideal and the worst of the related objective.

Step 3. After the goals are entered, the method then offers a subset of solutions closest to the goals for decision maker's consideration. Users are asked to select a satisfactory solution by clicking the ***Select solution*** button or to revise the goals by click the ***Revise goal*** button (Figure 7-8(3)). If the decision maker finds a satisfactory solution the method stops, otherwise it again interacts with the decision maker for revising the goals.

Step 4. If ***Select solution*** button is clicked, users can double click one solution from the set of solutions in the table. This solution is displayed in detail as the final satisfactory solution and the ISGP stops.

Step 5. If the decision makers want to change preferences for the objectives, they can revise the goals for the objectives. After the ***Revise goal*** button is clicked, the goal input table with the ideal values and worst values is displayed again. Step 3, 4 and 5 is repeated and the second set of solutions is obtained.

Step 6. This iterative process continues until the final satisfactory solution is found (Figure 7-8(4)).

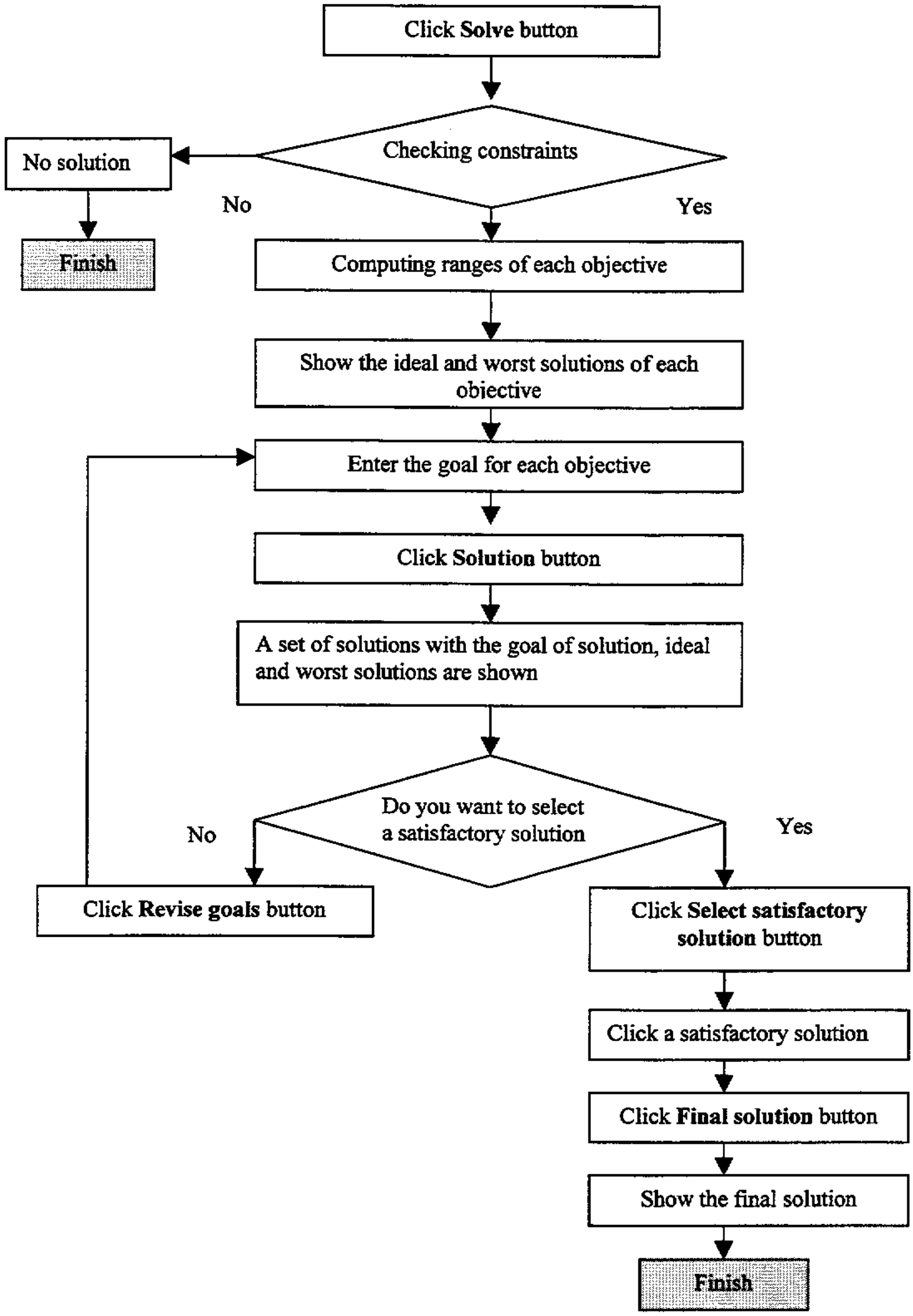


Figure 7-7 Algorithm of ISGP

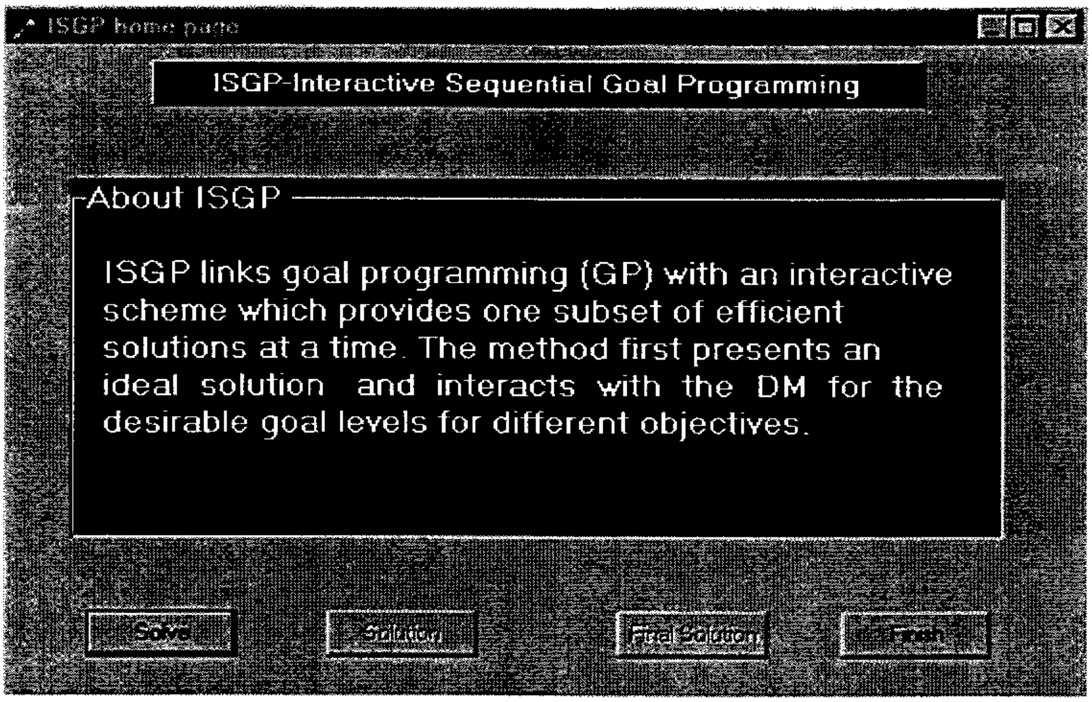


Figure 7-8(1) Home page of ISGP method

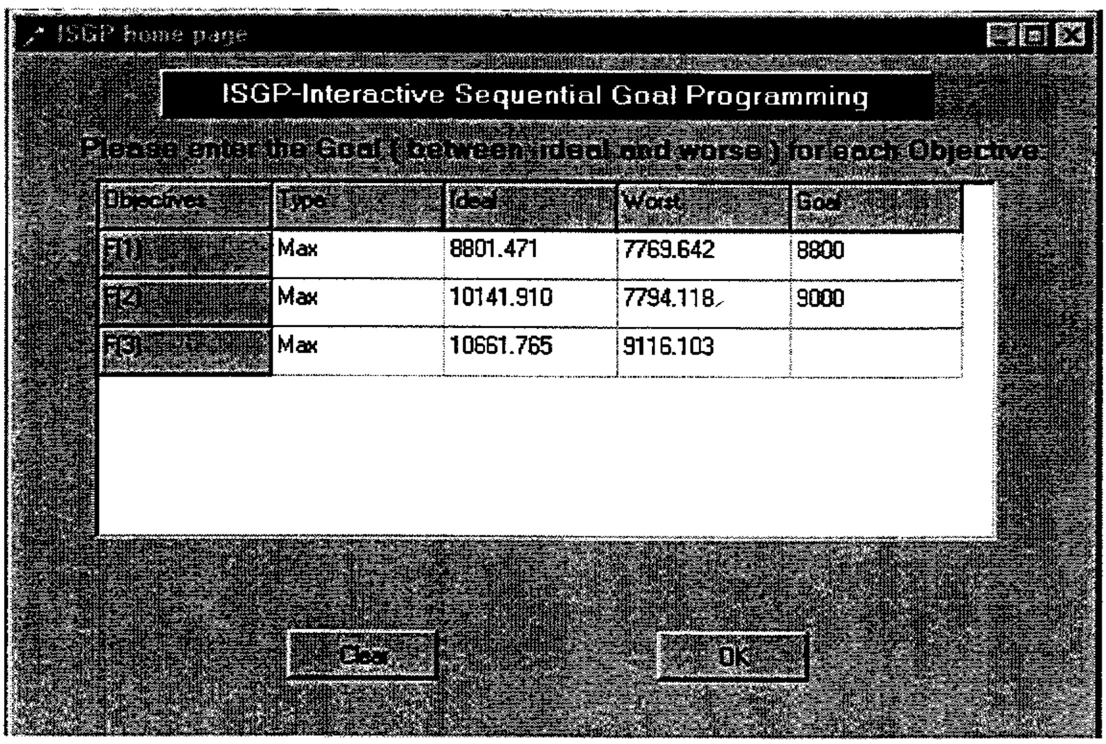


Figure 7-8(2) Goals input

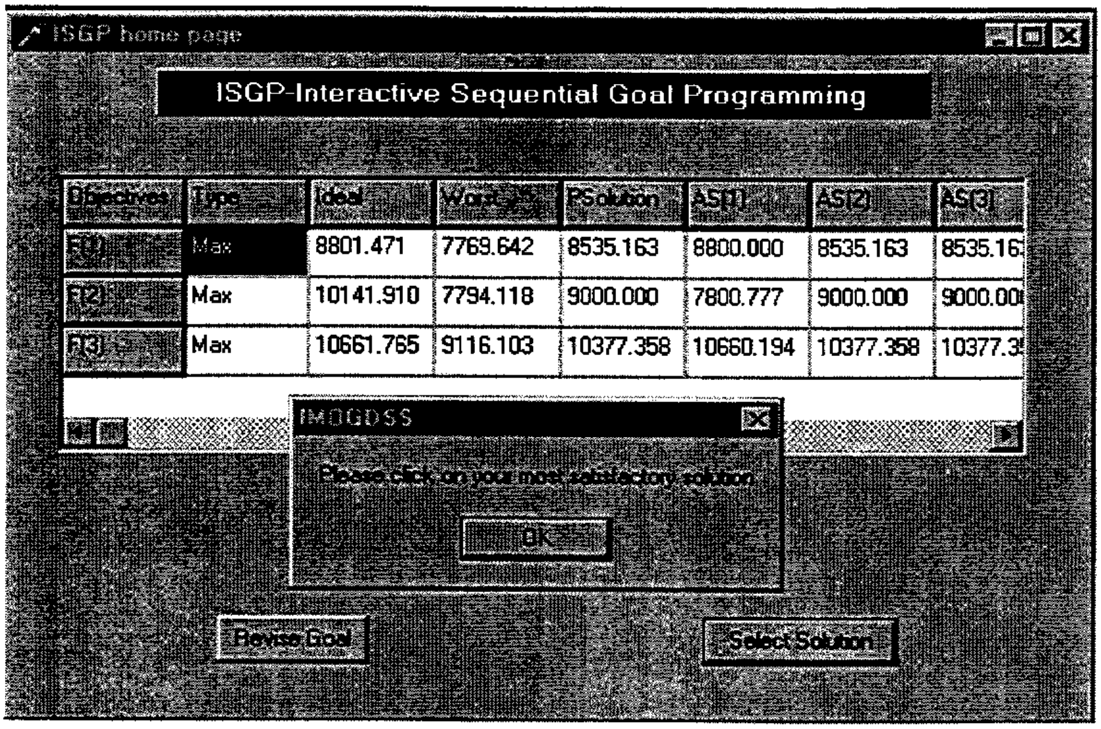


Figure 7-8(3) Select a satisfactory solution by double-click

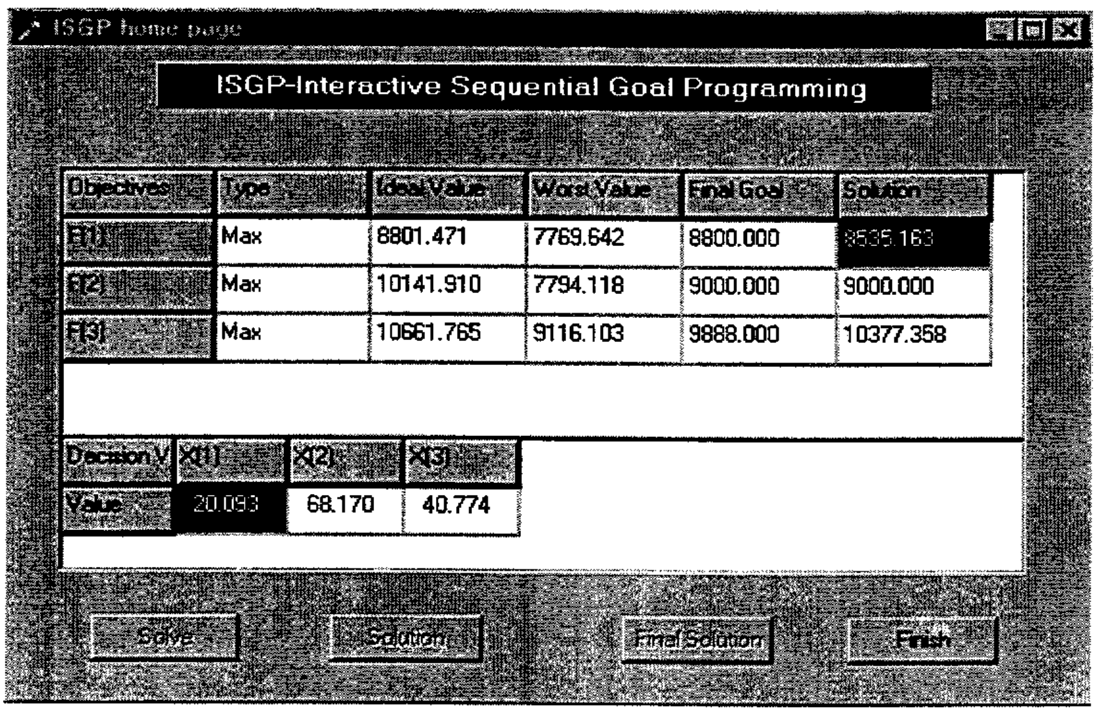


Figure 7-8(4) A final solution is obtained

7.6 Linear Goal Programming (LGP) Method

LGP (Ignizio 1976) is one of the oldest methods for solving MODM. It is non-interactive (non-pro-interaction in our definition), but has been widely applied to a variety of problems. In LGP the objective functions are transformed into the constraints, priorities are set for each objective, and the undesirable deviations from the goals are minimized. The method generates a unique solution which is not necessary an efficient one. It should be mentioned that different variations of goal programming are now available which eliminates the disadvantages of the basic version of goal programming. LGP has the following features:

- Users need to enter goal, weight and priority for each objective
- Generates a set of alternative solutions
- Allows users to select a satisfactory solution from the solution set

Figure 7-9 is the algorithm of LGP method. The performance process in IMOGDSS is described as follows:

Step 0. Click LGP menu to show the home page of LGP.

Step 1. Press ***Goal*** button to enter the goals, priorities, weights for objectives (Figure 7-10(1)).

Step 2. Click ***Solve*** button, the system starts to compute the ranges of each objective.

Step 3. After ***Solution*** button is clicked, the method offers a solution closest to the goals for decision maker's consideration. Users are asked to select the solution as a satisfactory solution, or to click ***Alt-solution*** button to see more alternative solutions (Figure 7-10(2)). If the decision maker is satisfied with the solution the method stops, otherwise it again interacts with the decision maker for the set of alternative solutions.

Step 4. Once ***Alt-solution*** button is clicked, a new solution is obtained and the message 'are you satisfied with this solution' is shown.

Step 5. If the decision maker accepts this solution the LGP method stops, otherwise, another new solution is shown.

Step 6. This process continues until the final satisfactory solution is found (or number of alternative solutions are exhausted).

This method requires more information from the decision makers. It is suitable for the decision maker who has prepared these data.

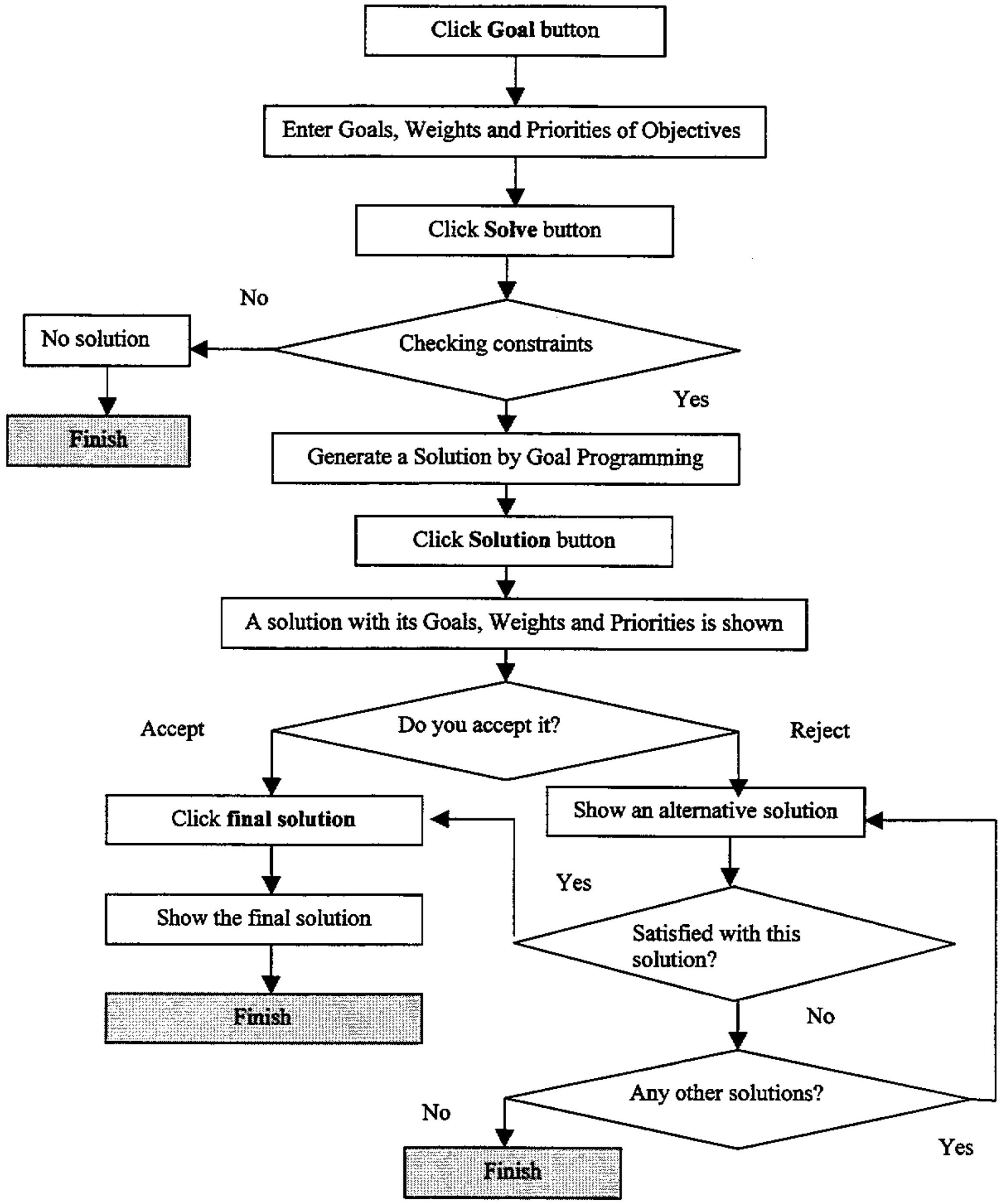


Figure 7-9 Algorithm of LGP method

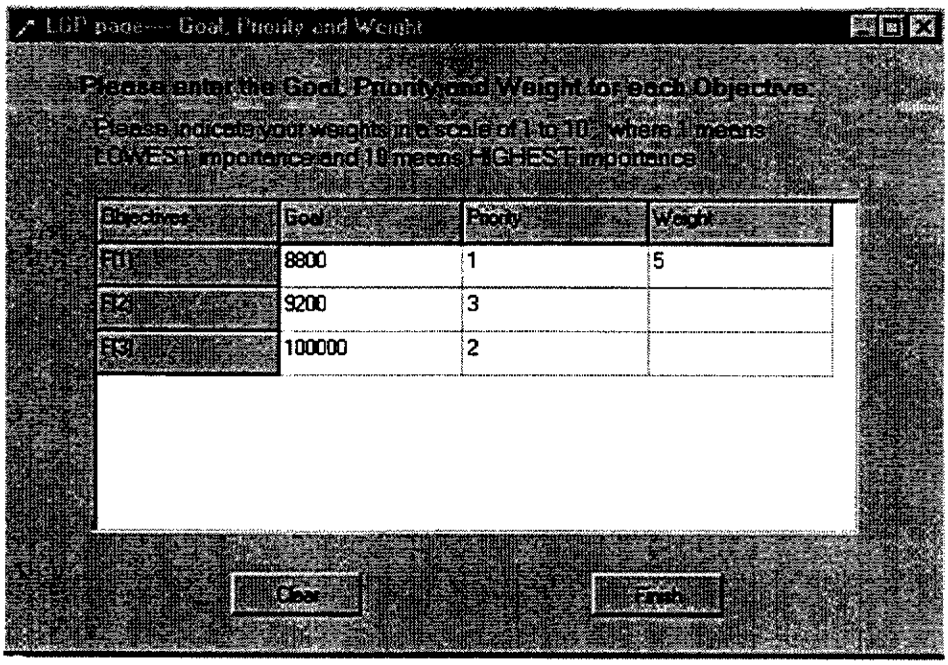


Figure 7-10(1) Enter the goal, priority and weight of each objective

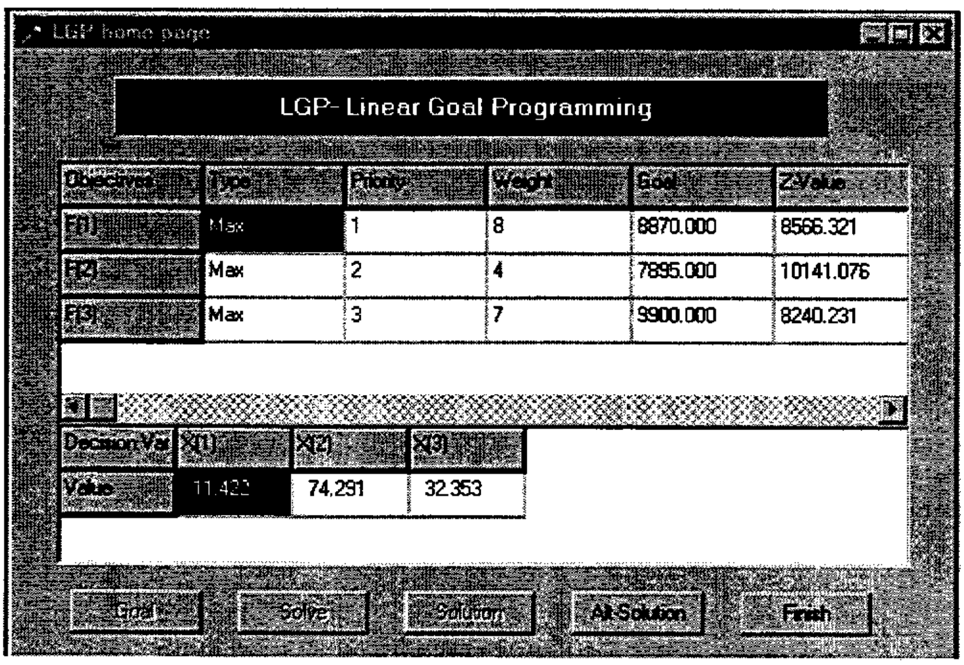


Figure 7-10(2) Press alt-solution button to browse alternative solutions

7.7 Step Method (STEM)

STEM (Benayoun et al. 1971) is one of the oldest methods to solve MOLP. It is interactive in nature. The method starts with finding a solution closest to the ideal solution in the MAX/MIN sense. The solution is then offered to the decision maker who is required to relax one of his satisfactory objectives so as to allow improvements in other objectives. The method then finds a new solution and interaction with the decision maker continues. The Algorithm is described in Figure 7-11.

STEM allows the decision makers to learn to recognise good solutions and the relative importance of the objectives. STEM has the following characteristics:

- Provides an ideal solution
- Provides a relaxation process for users to explore more solutions
- Generates a set of solutions through implicit trade-off
- Allows users to select a satisfactory solution from the solution set
- Allows users to make multiple pro-interaction and post-interaction

In the method, phases of computation alternate (interactively) with phases of decision. That is, (1)-(2)-(3)-(2)-(3)-(2)...in the following steps:

(1) construction of a pay-off table: a pay-off table is constructed before the first interactive cycle.

(2) calculation phase: the feasible solution to linear programming sought which is the nearest, in the MIN/MAX sense, to the ideal solution.

(3) Decision phase: the compromise solution is presented to the decision maker, who compares its objective vector with the ideal one. If some of the objectives are satisfactory and others are not, the decision maker must relax a satisfactory

objective enough to allow an improvement of unsatisfactory objectives in the next iterative cycle. Then, for the next cycle the feasible region is modified.

The STEM process implemented in IMOGDSS follows these steps:

Step 1. After the "home page" is shown, click ***Solve*** button, the system examines the consistency of the constraints and forms first solution.

Step 2. Click ***Solution*** button, a solution and the ideal value of each objective are displayed. Users are asked to accept or reject the solution (see Figure 7-12(1)).

Step 3. If ***Accept*** button is clicked, then this solution becomes the final satisfactory solution and the STEM method stops. However the decision makers have different preferences and different favourite choices, and they often like to further search so that more solutions are obtained. The ***Reject*** button is thus clicked and a relaxation process is invoked.

Step 4. After the ***Reject*** button is clicked, users are told to have to accept a certain amount of relaxation of a satisfactory objective to allow an improvement of the unsatisfactory ones in the next cycle. If users agree to this, they can start to relax the values of objectives under a guidance of the system. To decrease the value of an objective, enter a positive adjustment value otherwise enter a negative adjustment value to increase the value of an objective (Figure 7-12(2)).

Step 5. In some cases the decision makers' relaxation values are not feasible. When a relaxation fails, the system will enable the users to continue to re-enter a new set of relaxation values.

Step 6. A new solution will be displayed in the second cycle after users complete a new relaxation. Users can also accept or reject this solution. If the decision makers accept it, it is taken as the final satisfactory solution, otherwise the system repeats the above-mentioned relaxation process.

This method needs few interactions with users to arrive at a satisfactory solution, and it is therefore an easy to use method.

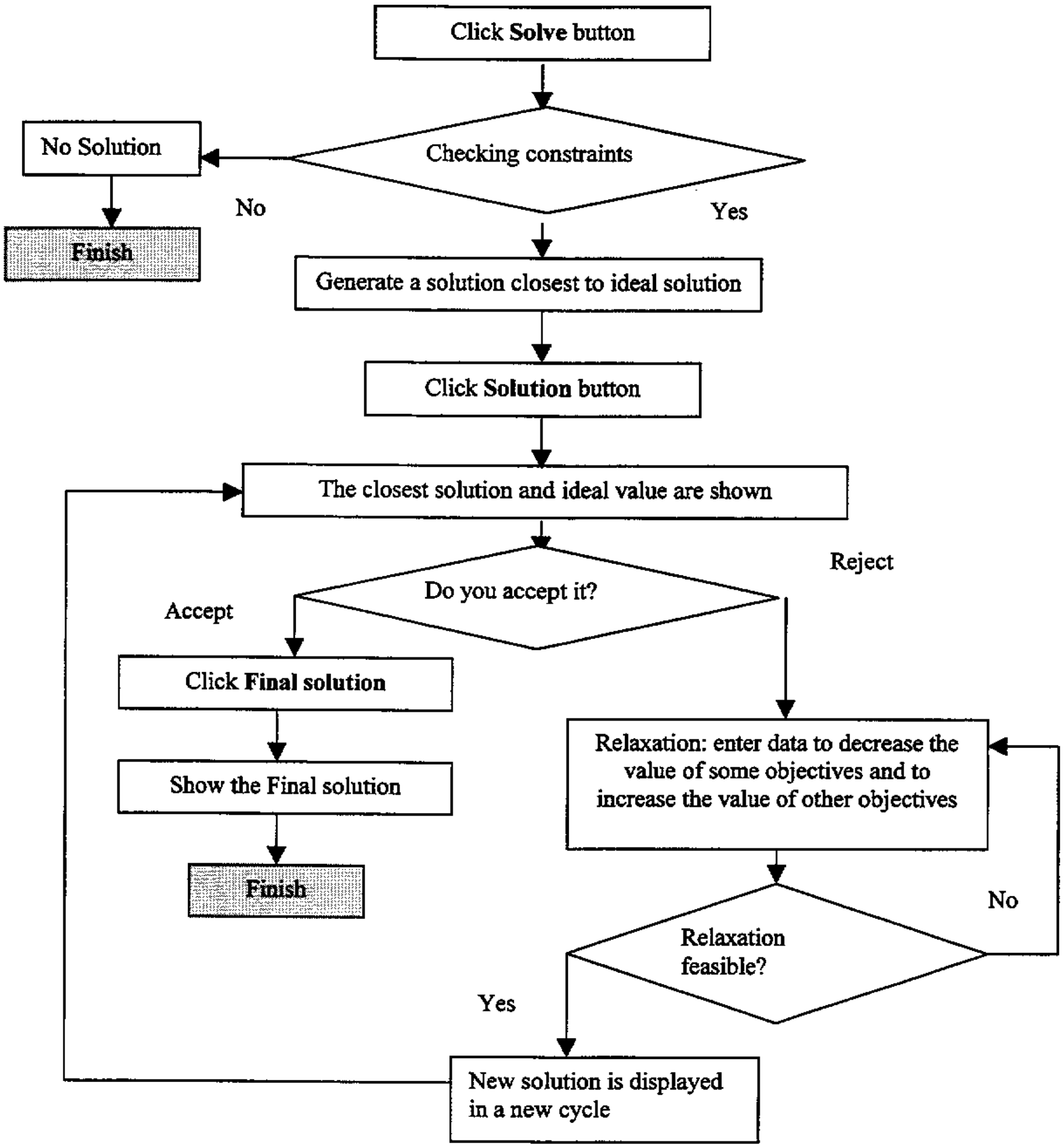


Figure 7-11 Algorithm of STEM

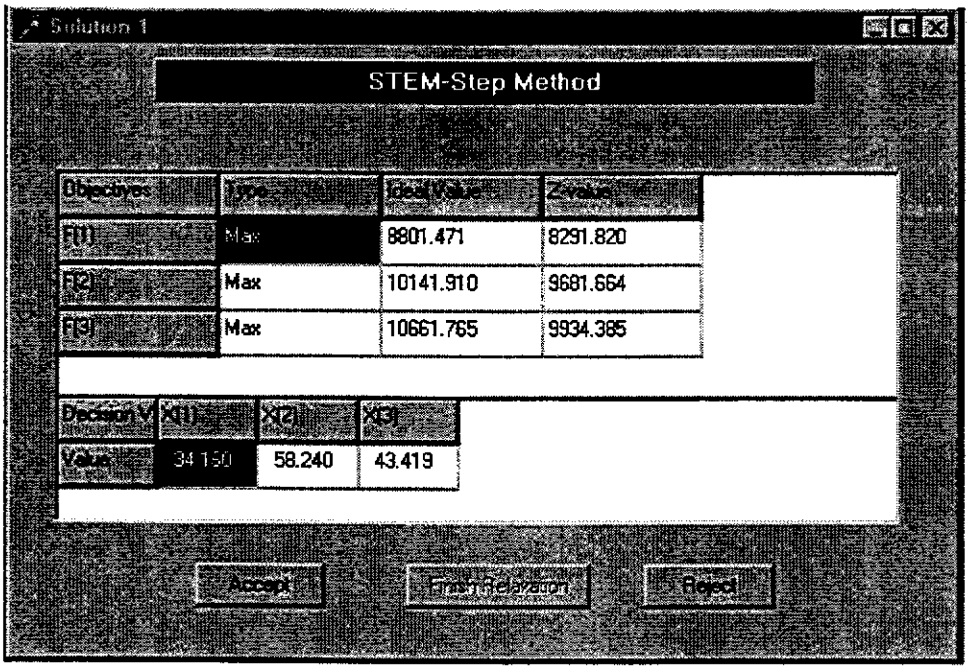


Figure 7-12(1) Users are asked to accept or reject the solution

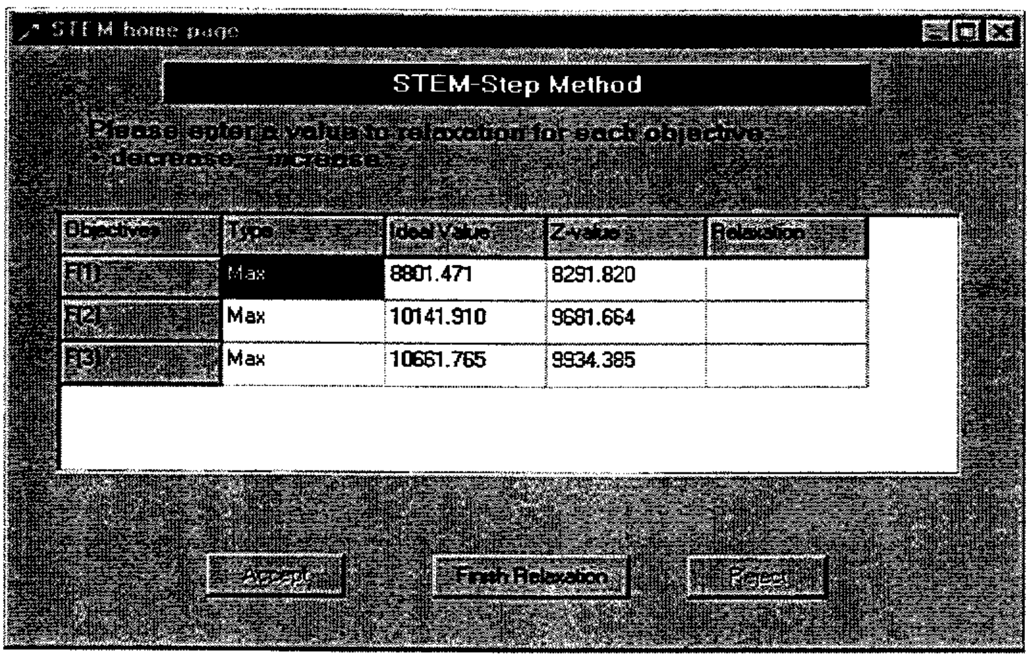


Figure 7-12(2) Decrease the value of an objective and increase the value of other objectives

7.8 STEUER Method

STEUER (Steuer 1977) is an interactive method to solve MOLP. It starts with presenting to the decision maker (2K+1) dispersed efficient solutions, where K is the number of objectives. The decision maker has to choose one solution from this set. The decision maker can then terminate the solution process or explore further (2k+1) solutions around the present solution. The method stops whenever the decision maker has found the most satisfactory solution. The implementation of STEUER is shown in Figure 7-13.

Summary, STEUER has the following features:

- Generats 2*k+1 solutions through implicit trade-off
- Allows users to select a satisfactory solution from the solution set
- Provides users to make post-interaction

In IMOGDSS, the decision makers can follow the following steps to obtain a satisfactory solution:

Step 1. Click ***Solve*** button, the system starts to find efficient solution. After the execution is completed, the system has computed dispersed efficient extreme points.

Step 2. Click ***Find Efficient Solution*** button, the system will generate and show 2*K+1 efficient solutions (k is number of objectives).

Step 3. Press ***Select Best Solution***, you may browse these solutions and indicate which solution is your most preferred. When you browse these solutions, there are three button shown: ***Previous***, ***Next*** and ***OK***. Button ***Previous*** is used to go back to the last solution and ***Next*** is used to go down to the next solution (Figure 7-14(1)).

Step 4. After the ***OK*** button is chosen, the system displays " Are you satisfied with the above value?" to confirm the users' selection.

Step 5. If ***Yes*** button is pressed, this solution is displayed in detail as the final solution and the STEUER stops. If button ***Not*** is chosen, the solution browsing process continues. Users can press ***Previous*** button and ***Next*** button to browse these solutions. The method checks each solution with the users.

Step 6. If users want to determine all efficient extreme points with respect to the current gradient cone, press ***Explore Further*** button until a satisfactory solution is found (Figure 7-14(2)).

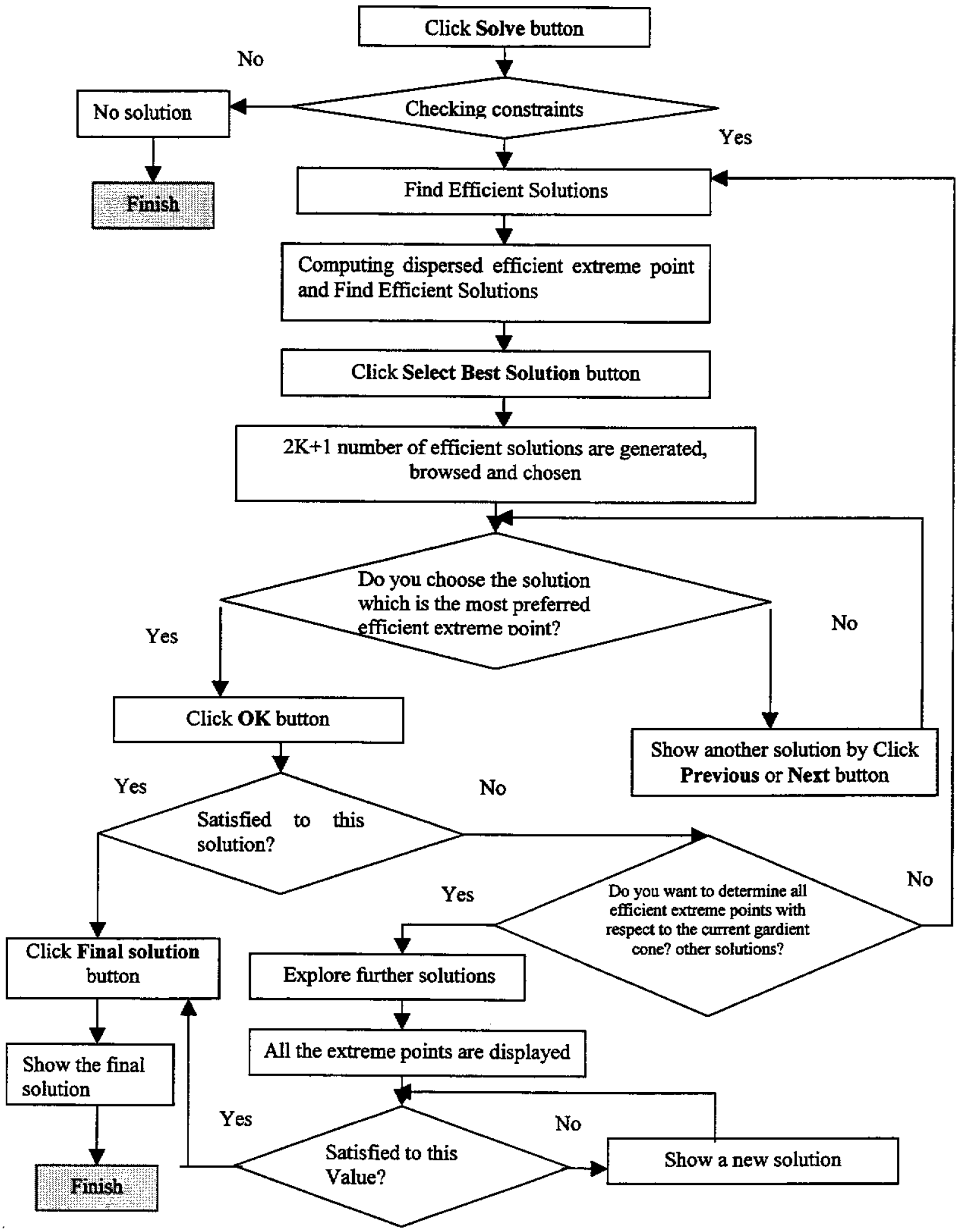


Figure 7-13 Algorithm of STEUER

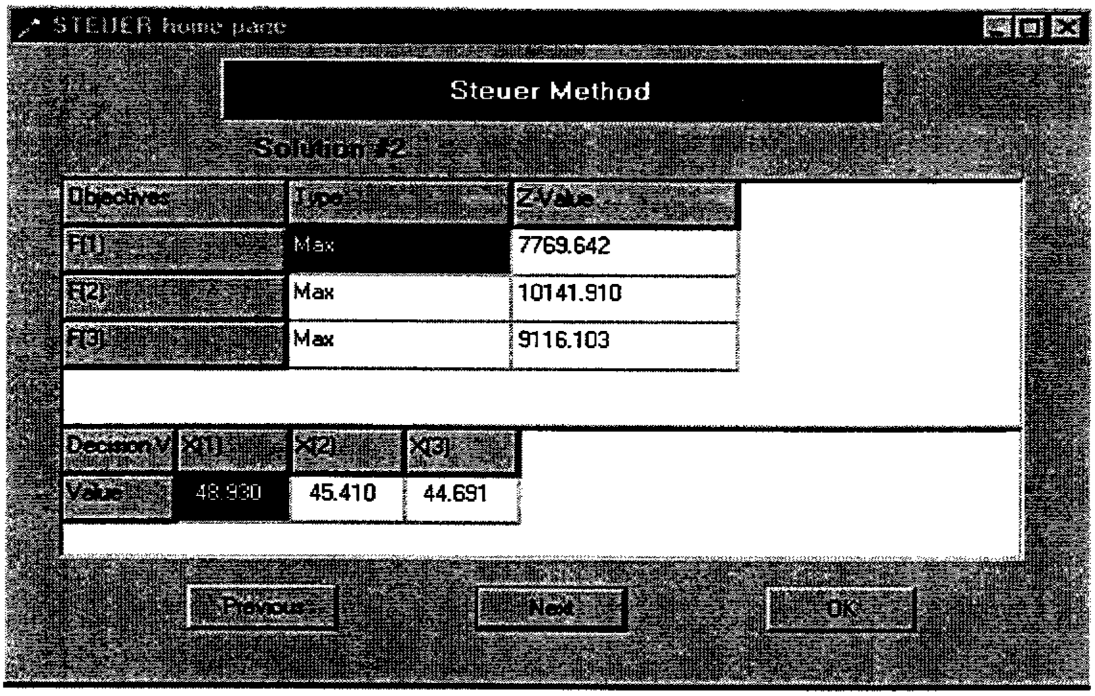


Figure 7-14(1) Browsing these solutions

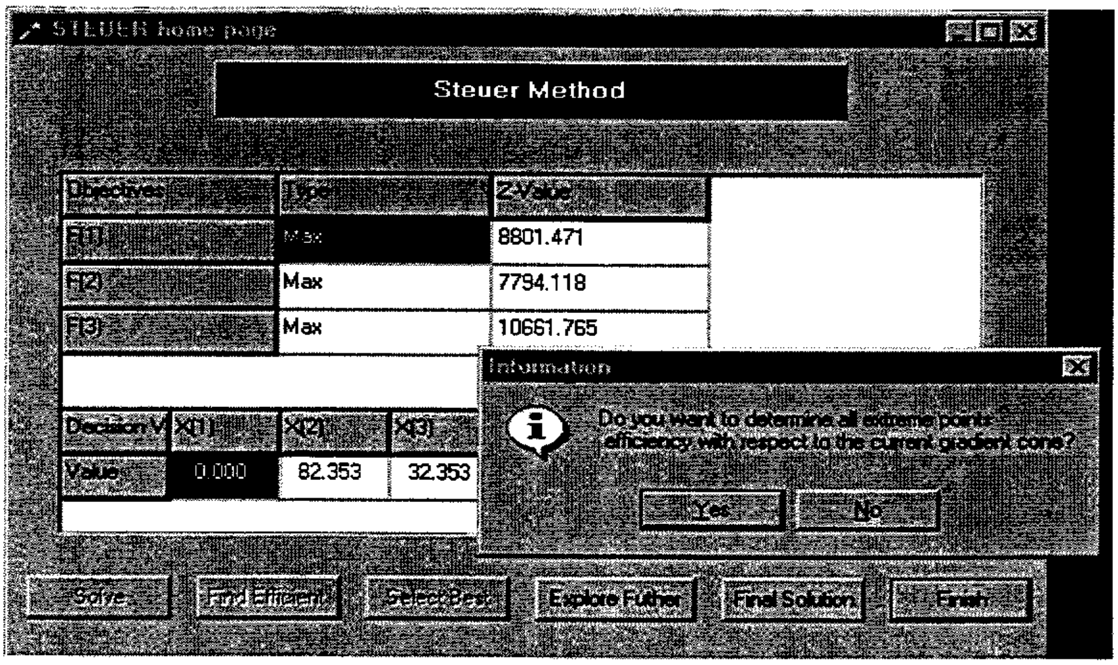


Figure 7-14(2) Explore to determine all efficient extreme points

7.9 Zionts and Wallenius (ZW) Method

ZW Method (Zionts & Wallenius 1976) is assumed that all the objective functions are linear (to be maximised) and the constraints form a convex set. The overall utility function is assumed to be unknown explicitly to the decision maker, but is implicitly a linear function and, more generally, a concave function of the objective functions. The method makes use of such an implicit function on an interactive basis. The algorithm of ZW is described as follows:

(1) Initialisation: choose an arbitrary set of positive multipliers or weights initially

(2) Generate and solve the composite objective function

(3) Find a set of efficient variables

(4) Decision phase: for each of the efficient variables, the decision maker is asked to accept a trade off in objective 1 to k, by responding *yes, no* or *indifferent* to the trade off. If the responses are all no for all efficient variables, the procedure is terminated, and current multipliers are taken as the best set of weights. Otherwise, for each *yes, no*, and *indifferent* response an inequality is formed.

(5) Find the new set of weights: a feasible solution is found from the inequalities of step 4. The method has goes back to step 2.

(6) The method is either terminated by the decision maker or by the process itself after finding a unique solution (or no solution).

ZW is implemented in IMOGDSS as shown in Figure 7-15. Decision makers can follow the following steps to find a satisfactory solution:

Step 1. After the home page of ZW is shown, click ***Solve*** button, the system starts with an arbitrary efficient solution.

Step 2. Click ***Solution*** button, this system then interacts with the decision maker. A solution and the weight value of each objective are displayed. The users are asked to accept or reject the solution (Figure 7-16(1)).

Step 3. If ***Accept*** button is clicked, then this solution becomes the final satisfactory solution and the ZW stops. However the decision makers have different preferences and different favourite choices, and they often like to further search so that more solutions are obtained. The ***Reject*** button is thus clicked. The system presents a set of specific trade-offs.

Step 4. After the ***Reject*** button is clicked, a set of decrease/increase value trade-offs is given. The users are asked " Are you willing to accept these relative trade-offs?" and ***Yes, No*** and ***Ignore (Indifferent)*** buttons are shown. The decision maker has to answer "Yes", "No" or "Indifferent". Figure 7-16(2)) shows this step.

Step 5. If users choose ***Yes***, that means they agree to the set of decrease and increase values, so a new solution is obtained. Step 1 to Step 4 are then repeated. If users choose ***No***, the system will change to another group of decrease/increase values. Step 4 is then repeated. If users choose ***Ignore***, a message is shown " your present trade-offs are inconsistent with your previous information, do you want to revise?". You can select ***Yes*** or ***No*** button. System thus goes back to step 4.

Step 6. After users choose ***Yes*** for the second set of decreasing /increasing values, the second solution is found and displayed (Figure 7-16(3)). Users can accept or reject this solution. If the decision makers accept it, it is determined as the final satisfactory solution, otherwise the system repeats the above-mentioned relaxation process.

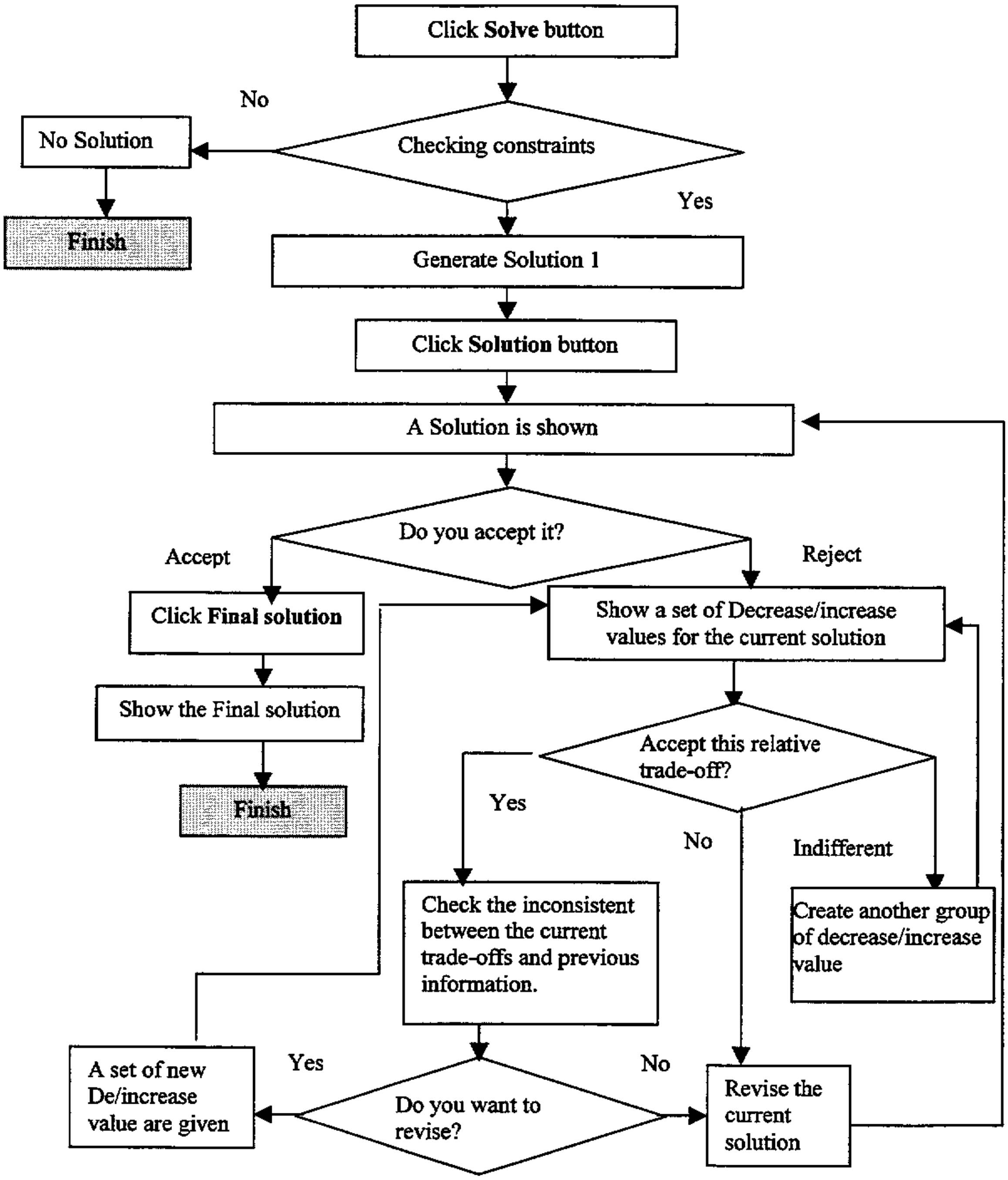


Figure 7-15 Algorithm of ZW

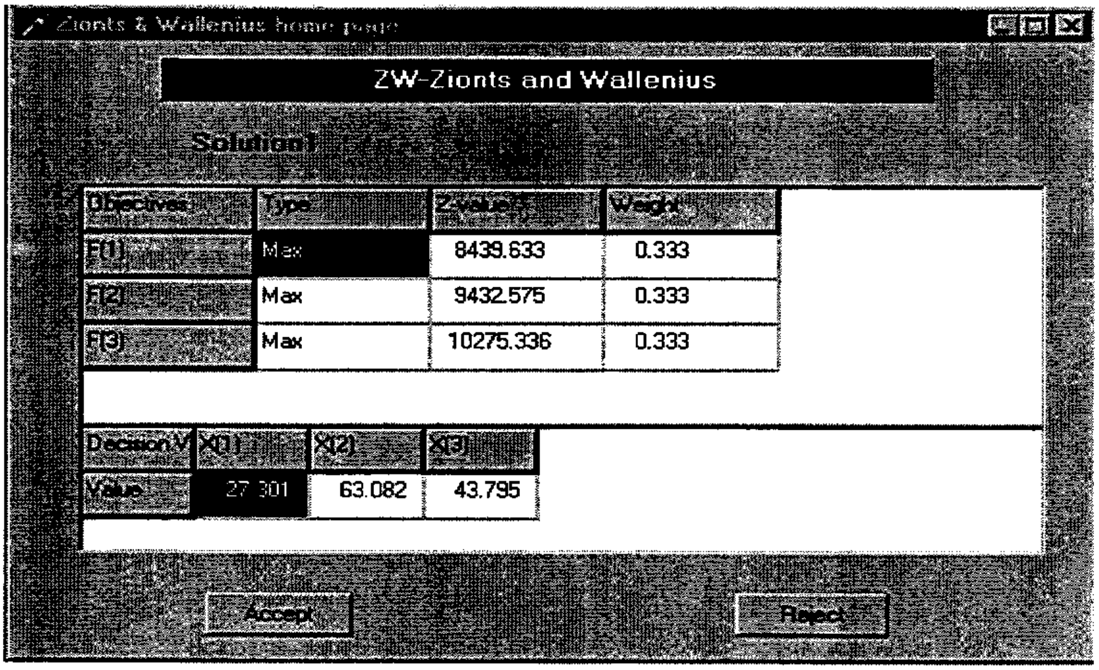


Figure 7-16(1) Users are asked to accept or reject a solution

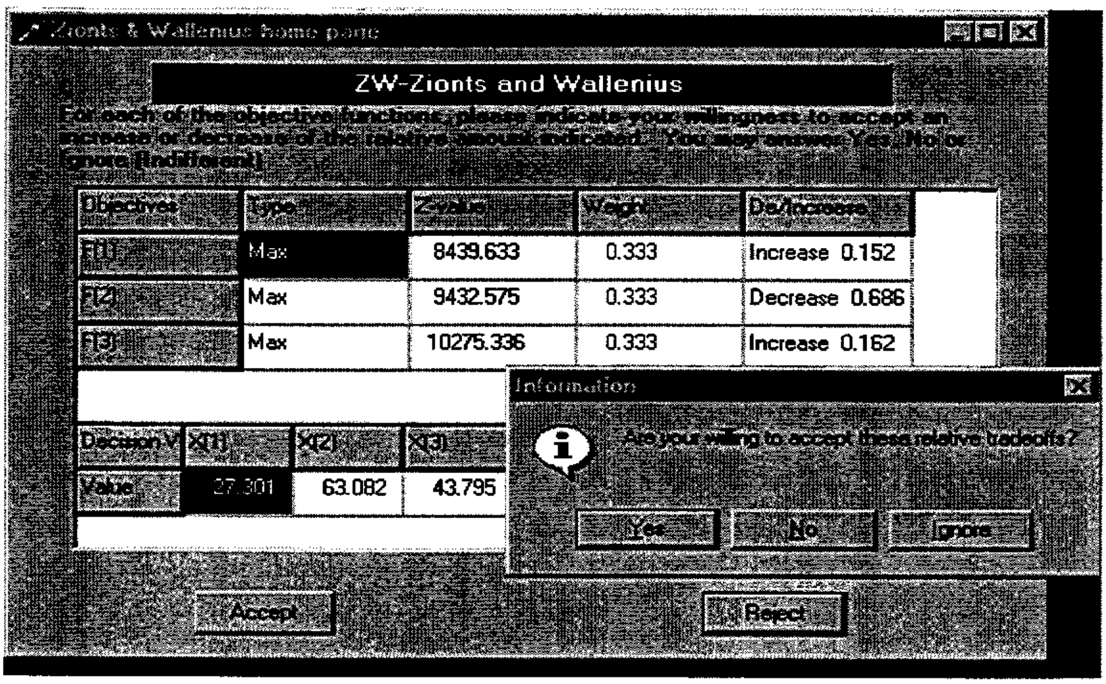


Figure 7-16(2) Are you willing to accept these relative trade-offs?

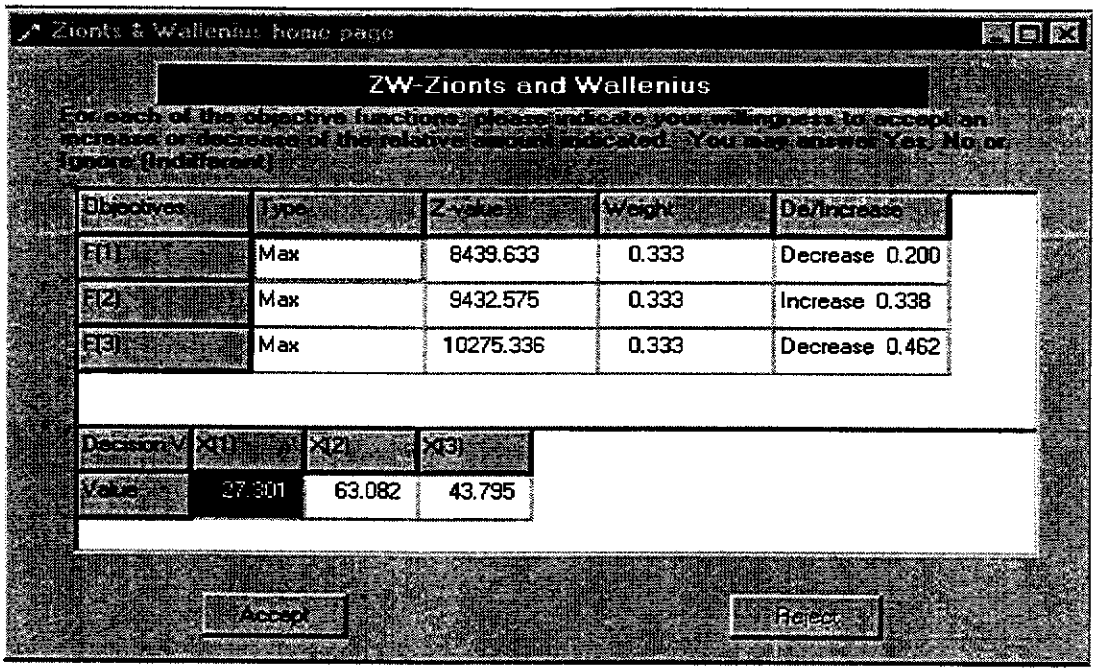


Figure 7-16(3) Accept this solution

7.10 Summary

As a main component of IMOGDSS, MODM methodology base is used to provide MODM problem solution through various types of interactive processes. This chapter proposes the architecture and features of MODM methodology base, displays its data structure, interaction types and central role in IMOGDSS for supporting decision making.

This chapter illustrated the design and implementation of the MODM methodology base, described the working process of seven MODM methods and exhibits that these methods have different characteristics and they will be suitable for different decision makers. The solution process provided in MODM methodology base is friendly and easy to understand, the methodology base is effective and GUI-based, the method subsystem is flexible and it is used to enhance the quality of decision making. This chapter has answered the research question 1.1 with respect to developing MODM methodology base.

Chapter 8

MODM-BASED GROUP DECISION SUPPORT, GROUP AGGREGATION METHODOLOGY BASE AND GROUP SUBSYSTEM'S IMPLEMENTATION[6]

8.1 Introduction

Multiple objective decisions are often taken in a decision group environment. Therefore, MOGDSS should be built to provide a group of decision makers with feedback to individual preferences regarding possible solutions to MODM problems and make an aggregation for all group members' solutions. Usually, different group aggregation methods are suitable for different types of group decision making and group meetings. Integrating multiple group aggregation methods into a system will be very useful to support a wide range of group decision situations, to find the most appropriate method to solve specific decision problem, and to produce a final compromise solution through multiple methods.

[6] This chapter is based on the following published paper:
(1)*Lu, J., M.A. Quaddus and R. Williams, "Intelligent multiple objective group decision support systems: a review, prototype and an application", Proceedings of Acer National Business Education and Research Conference, 14-15 October, 1999.*

This chapter proposes a framework for such requirements through imbedding a group component (subsystem) in IMOGDSS. The group subsystem has been designed and implemented to include a group decision making (GDM) methodology base (or group aggregation methodology base) to aggregate a group of MODM solutions as alternatives, that involve multiple conflicting objectives and are provided by multiple decision makers with different business functions and preferences. This GDM methodology base consists of five group aggregation methods and allows the group members to select any aggregation method. A best compromise solution is then obtained through an interactive/non-interactive procedure.

The technology of aggregating MODM in GDSS has received little research attention, and the implementation of MOGDSS to generate alternative solutions through running an MODM methodology base and to produce a compromise solution through running a GDM methodology base has not been found in the literature. The IMOGDSS provides a new configuration of group decision support software.

This chapter provides an answer for the research question 1.3 with respect to designing and implementing group methodology base in order to find a compromise solution for the group.

This chapter begins with an introduction to the purpose of MOGDSS and group aggregation in this study. Next, Section 8.2 briefly summarizes the goals, features of the group subsystem and the affected factors for the performance of group aggregation and group decision support. Section 8.3 describes the framework of MODM-based group aggregation and construction architecture of group subsystem in IMOGDSS. Five group aggregation methods involved in the GDM methodology base are outlined in Section 8.4. Section 8.5 discusses the design, implementation and application of the group subsystem in IMOGDSS. The summary is presented in Section 8.6.

8.2 Goals, Affected Factors and Features of Group Decision Support

8.2.1 Goals of Group Decision Support

Systems that combine appropriate technologies, methodologies and facilities of DSS and GDM, show the potential for providing such flexibility to enhance the efficiency and effectiveness of group decision work. Such applications of information technology to support the decision work of groups have been referred to as group decision support systems (GDSS) (Gray 1987). GDSS combine computer, communications, and decision technologies to support decision making problem finding, formulation, and solution in decision group meetings (DeSanctis & Gallupe 1987). Because decision making in a group occurs through interpersonal communication, the most fundamental goal of a GDSS is to support the exchange of ideas, opinions, and preferences within the decision group and find a compromise solution for the decision problem (Gallupe, DeSanctis & Dickson 1988).

Prior research on GDSS exhibits a number of characteristics in common. Typically, the tasks used are idea generation and choice tasks. Members of the group have the same stake in the outcome. Group members have had little or no prior interaction as a group and will have little or no interaction after the exercise. With few exceptions, the group studied do not have a formal leader (Barkhi et al. 1998). This study broadens the context of a GDSS and builds a group subsystem that suits with broad characteristics of a group environment. In an organization, decision groups often confront complex tasks that involve multiple objectives and do not have the best solution. Increasingly, teams and groups are geographically dispersed. The goals of each group member may differ. Not all members will have the same information (information asymmetry). Often, group participants play a different weighted role for a decision solution. Finally, groups may or may not have a formally appointed leader (group authority) (Barkhi et al. 1998). It has been found that any change in each of these characteristics will affect the performance of the group meeting. Also, some group aggregation methods are more suitable than others in the performance of some group decision making, which posses some of the specific characteristics mentioned above.

8.2.2 Affected Factors of Group Performance

The performance of group decision processes can be affected by five main factors:

- participant information factor;
- communication channel factor;
- group leadership (group membership weighting) factor;
- decision making time factor; and
- decision making place factor.

Each of them has two modes. For example, participants of the group do not have the same information (i.e., goals) sometimes, because these group members represent a different business function (e.g., marketing, production, purchasing) and have conflicting objectives in a decision problem. The communication channel is an important factor and its mode will affect other factors. There are two communication channel modes: face-to-face (FTF) communication and computer mediated communication (CMC). FTF mode is only suitable in a decision meeting that is arranged at the same time and same place. CMC can be used for a decision meeting which is held at the different time and different places. CMC also may promote de-individuation by reducing the number of channels that are used for personal interaction (Barkhi et al. 1998). Leadership and its influence on small groups have been studied and leadership has not received wide attention in the GDSS literature (Barkhi et al. 1998). The leader (or leaders, or few members) acts as a cohesive force, helps select an aggregation method and achieve task-oriented goals, and improves the level of consensus by trying to be fair to every member. As a broad consideration, the role of these group members' feedback also can affect the result of decision meeting in different important degrees, that is, some members' preferences are more important than others. Anyway, for situations in which the knowledge necessary to solve a problem is distributed among different group members (Barkhi et al. 1998), the importance of the group members' preference that is aggregated should be allowed to be different. These are the five main affect factors considered in this study. Some GDSS researches have also focused on the role and impact of an independent facilitator (Niederman, Beise & Beranek 1996). However, this facilitator doesn't play any role in the determination of a final solution except that a series of facilitated operations is completed.

Based on above analysis, a 5×2 factorial design type comprehensive factors are summarised below:

1. ***Participant information factor***

   Mode 1: I(a)--the participants of the group do have the same information

   Mode 2: I(b)--the participants of the group do not have the same information

2. ***Communication channel factor***

   Mode 1: C(a)-- face to face (FTF) communication

   Mode 2: C(b)-- computer mediated communication (CMC)

3. ***Group leadership mode***

   Mode 1: G(a)--decision group with a leader or with different weightings for members

   Mode 2: G(b)--decision group without a leader and all members have same weight

4. ***Time mode of decision making***

   Mode 1: T(a)--group meeting at the same time

   Mode 2: T(b)--group meeting at the different time (work asynchronously)

5. ***Place mode of decision making***

Mode 1: P(a)--group meeting at the same place

Mode 2: P(b)--group meeting at the different places (distributed communications environment or internet via environment)

The above factors and their modes will be used later in connection with group aggregation methods.

8.2.3 Features of Group Aggregation

Recent research by Gray & Mandviwalla (1999) indicated that GDSS research has reached the point where we need to expand what we can do with GDSS. Growth can come in:

- Capabilities: increasing within group support so that more group activities are supported during a meeting
- Range: support more organizational tasks
- Effectiveness: improve the effectiveness of groups

"To do so, we will need to invent, invent, invent rather than test, test, test what exists" (Gray & Mandviwalla 1999). One of the needs is to invent new ways of using software to support the whole procedure of a decision meeting and support various kinds of decision meetings. Thus it will be possible for people with different roles and functions to use the same software and data in a meeting that they use on their desk, and group members are allowed to work asynchronously from the meeting.

GDSS is a field in which hardware and software are continually being upgraded and innovative approaches to group meetings can be developed (Gray & Mandviwalla 1999). The group subsystem provided in this study is trying to develop an innovative approach to group meetings and invent new methods for GDM. The main feature of the group subsystem is the incorporation of a group decision making methodology base

(GDMMB) that will provide a number of group aggregation methods for the different situations of groups. This group subsystem is built on the MODM methodology subsystem of IMOGDSS. It works by depending only on the input of individual solutions of MODM problem produced using an MODM methodology subsystem by the group members. All members are supported by the MODM methodology base to get an individual MODM solution through similar or different methods. Since different decision makers have different preferences and backgrounds, they may choose different methods to solve one specific MODM problem and obtain corresponding solution with different values. This GDM methodology base will help to unify the solutions from multiple decision makers and achieve more group effectiveness.

8.3 Framework of MODM-based Group Aggregation

8.3.1 Two-Stage Performance Framework

Maximization and satisfaction are two major frameworks for decision making (Kersten 1985). A combination of both may be used for MOGDM. Each decision maker takes into account their preferences or wants to obtain a 'maximal' objective solution using the most suitable method. They then submit their solutions and formulate the aspiration levels for each objective in a group, and a compromise satisfactory solution is determined from among the different solutions. IMOGDSS thus follows two stages to complete the group decision making process:

- Stage 1: each decision maker makes a decision for the MODM problem;
- Stage 2: decision makers negotiate so as to achieve a compromise decision.

The first stage has been discussed in Chapter 6. This chapter mainly describes the second stage. In the second, decision making groups are formed to exchange information and ideas, and to identify acceptable and desirable solutions. However, many times decision groups reach a dead end due to the differences among individual interpretations of the 'best compromise' solution. There is no rule for combining individual preferences into a group preference unless interpersonal comparison of "utilities" is allowed. Consequently, most utility group aggregation methods require explicit interpersonal comparisons of utility and follow a normative approach assuming that a group decision rule can be constructed by aggregating the utility functions of group members (Iz & Jelassi 1990).

The appropriateness of any decision method within an MOGDSS depends on the conditions of members, tasks and decision environment, and decision makers often have the requirements of saving time, expediency and resolution. In a multiple objectives group decision making, the preferences of the group members are expected to vary from each other and the members have obtained a group of solutions for an MODM problem. These solutions reflect the different preferences of group members. Consequently, determining the best alternative solution to a multiple objective problem requires aggregation of individual preferences to find the 'best' compromise solution. This is especially true for an interactive procedure that requires group members to generate alternative solutions.

A facilitator is needed in the second stage of the group decision making process. He/she operates the group subsystem to import the members' solutions that are received by e-mail, discs or hard copy into the database of the group subsystem, then operates one or more group aggregation methods in the GDM methodology base to arrive at a compromise solution. The facilitator has no influence on the final solution of a group. Figure 8-1 shows the two-stage group decision structure.

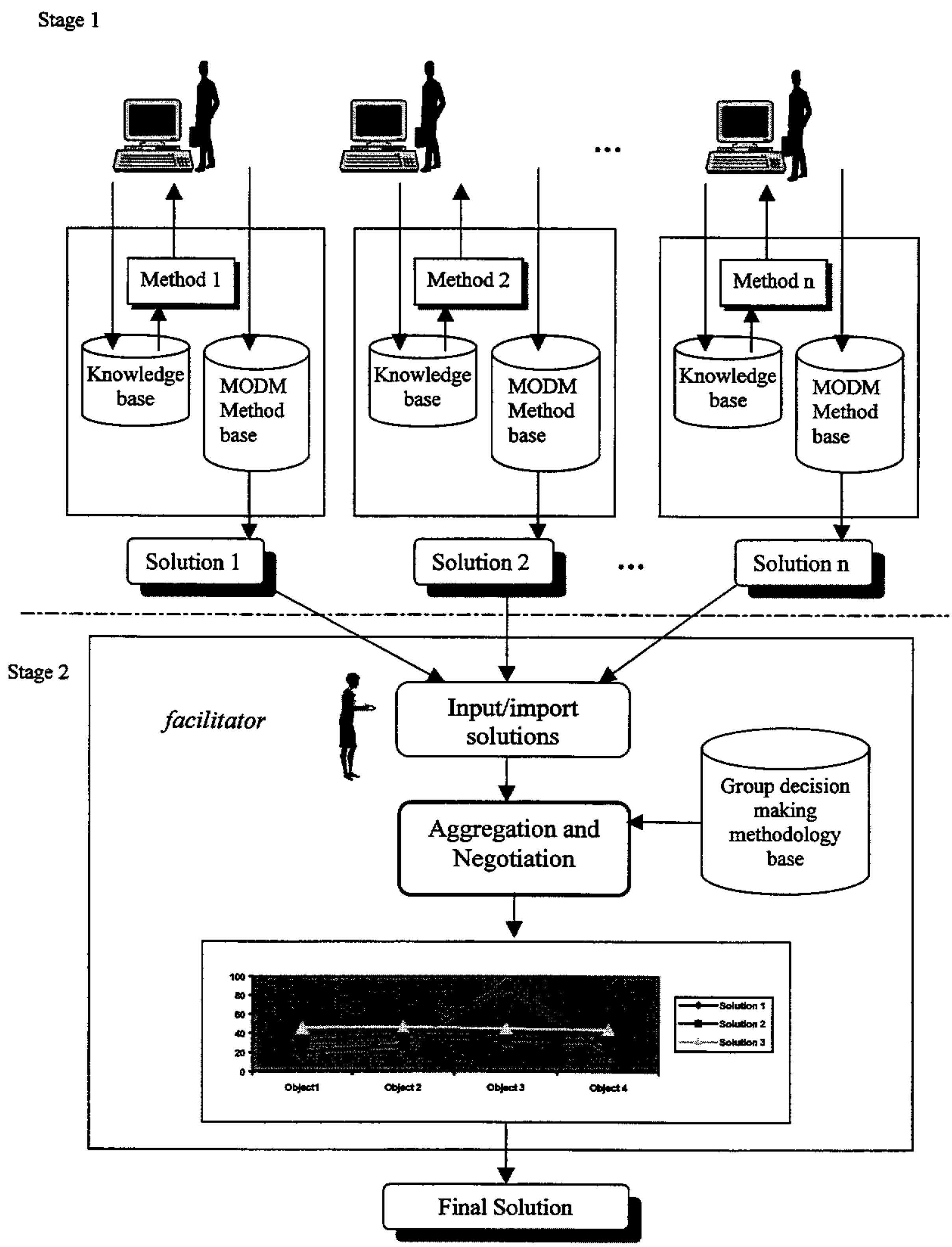


Figure 8-1 Two-stage structure of group subsystem

8.3.2 Construction of the Architecture of Group Subsystem

The group subsystem in the IMOGDSS consists of an input subsystem, an import subsystem, a GDM methodology base subsystem that includes five group aggregation methods, a data management subsystem and a report generation subsystem. Each subsystem uses graphical display techniques.

The input subsystem is used to input data by keyboard when a member's solution is provided in a hard copy. The import subsystem can read solutions of group members through electronic text files. These members' solutions are then converted into the form of group solution data file to save to a group database. The GDM methodology base aims to provide a set of GDM methods selected to support the group decision meetings that have different requirements for the solution procedure in order to solve different multiple objective decision problems. The group can try any GDM method in the GDM methodology base. Group database is used to store the group solution data and the report subsystem produces a group report. Figure 8-2 shows the architecture of the group subsystem.

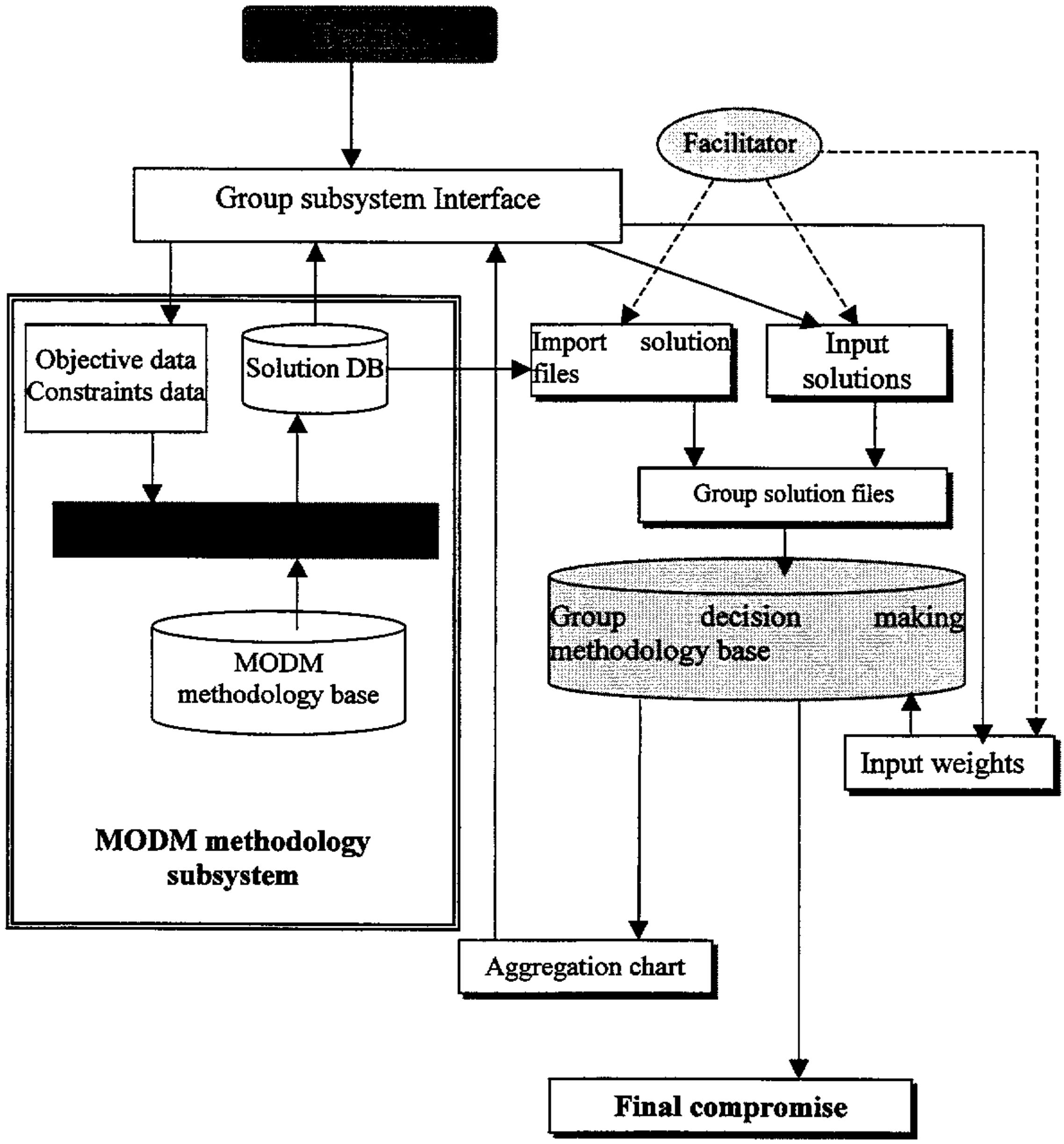


Figure 8-2 Architecture of group subsystem with a GDM methodology base

8.3.3 Functional Integration of Group Aggregation Methodology Base

It is the new configuration of the group decision support software that contains a number of GDM methods and integrates them in its methodology base for the selected usage during the group decision process. The objectives of integrating this methodology base are:

- to support a wide range of group decision situations;
- to help the group be more productive and more effective;
- to allow the use of multiple group decision methods for a group decision making tasks; and
- to find an appropriate method for every specific decision meeting situation and solve every specific decision problem.

Currently, five group aggregation methods are available in this GDM methodology base, some are based on interactive algorithms and some are non-interactive. However, the flexibility of the subsystem's modular structure will allow any new GDM method to be easily integrated in this methodology base. The focus of these methods is to determine a compromise solution to an MODM problem which best conforms to the preferences of the decision makers. These methods are implemented as independent executables to facilitate the flexibility required of the system. These methods share similar data acquisition routines and these routines are developed as independent modules so that data acquired could be accessed by all the methods. As common data structure routines, two matrices, called '*Objective matrix*' and '*Variable matrix*', are generated through the input subsystem or the import subsystem, and are described in the common data structure routines.

Many types of aggregation processes and related GDM methods have been proposed in the literature. However, they all work under a one-stage framework of GDSS. When the group subsystem is imbedded in IMOGDSS and a number of MODM solutions have been produced by different decision makers as alternatives, the GDM process is invoked to find the aggregate solusion. The objective of a group aggregation method is to choose a solution (alternative) that is the best compromise solution through an aggregation process. Figure 8-3 shows the five aggregation methods implemented in this GDM methodology base.

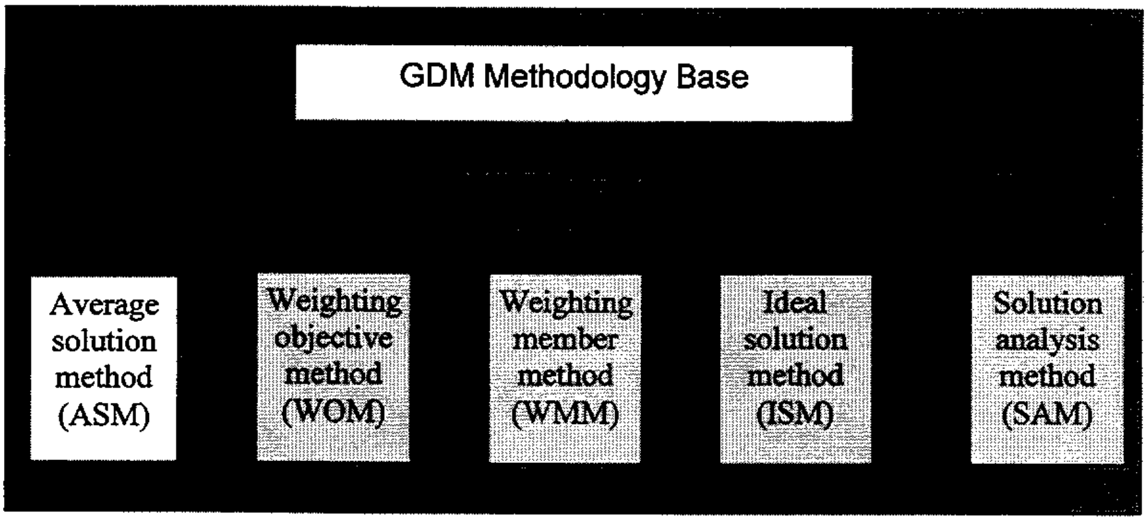


Figure 8-3 Group decision making (group aggregation) methodology base

8.3.4 Advantages of Interactive and Non-interactive GDM Methods

In this methodology base, some methods such as WOM involve an interaction, which has a number of advantages:

- it explores promising solutions rather than finding "the most compromised' solution in this group;
- it interacts with each problem owner and thus reflects his/her preference structure;
- it evaluates the alternative solutions, thus becomes a learning process for the group members who understand a great deal about the aggregation process;

- it allows the group members to efficiently explain and clarify their preferences and be given different weights;

- it can avoid the decision groups reaching a dead end due to the differences among individual interpretations of the 'best compromise' solution.

Some methods such as ASM involve a non-interactive approach which has a number of advantages:

- it is easy to use in a 'group decision meeting' where members provide the individual solution at different times and different places;

- a compromise solution can be quickly reached;

- it can be first selected and used to obtain a preliminary group solution, and another interactive method is then used to analyse this preliminary solution.

Table 8-1 shows the interactive or non-interactive characteristics of the five GDM methods.

Table 8-1 Interactive or non-interactive characteristics of GDM methods

Type	*ASM*	*WOM*	*WMM*	*ISM*	*SAM*
Interactive		*	*		*
Non-interactive	*			*	

8.3.5 A GDM Method-Factor Relationship Model

This subsystem can be applied to produce a compromise solution through many kinds of meeting processes between the problem owners and the system. A 'single method procedure' means only one GDM method is used in a group meeting, while a 'multiple method procedure' applies two or more GDM methods in a group meeting. In an interactive process, group members need to define clearly and input their weights for objectives or members, or make a solution analysis through a relaxation process. But in a non-interactive process, a final solution is displayed immediately once the group members' solutions are imported. A mixed process is the recommended approach. In this approach, choose a non-interactive method first to produce a preliminary compromise solution, and an interactive method is then selected to improve the solution until a satisfactory solution is found. Figure 8-4 shows several possible approaches to efficiently apply these methods in the methodology methods.

[redacted]	*Approaches*	[redacted]
	Single non-interactive method	
	Single interactive method	
	Multiple non-interactive methods	
	Multiple interactive methods	
	Mix interactive and non-interactive methods	

Figure 8-4 Combined applications of GDM methods

This subsystem presents the GDM methods under a similar framework for data import. The solution is displayed in a graphic format, and weight input, solution analysis and a satisfactory compromise solution is generated in order to facilitate the users to use this methodology base effectively.

In the GDM methodology base, each method is suitable for one or two modes of the five factors discussed in Section 8.2.2. Table 8-2 shows the relationships between these methods and the factors.

Table 8-2 The relationships between five methods and five factors

Suitable	I(a) Same data	I(b) Different data	C(a) FTF	C(b) CMC	G(a) Leader	G(b) No Leader	T(a) Same time	T(b) Different time	P(a) Same place	P(b) Different place
ASM	*	*	*	*		*	*	*	*	*
WOM	*	*	*		*	*	*		*	*
WMM	*	*	*		*	*	*		*	
ISM	*	*	*	*		*	*	*	*	*
SAM	*		*			*	*		*	

8.4 Group Aggregation Methodology Base

8.4.1 Average Solution Method (ASM)

The average solution method is also called the shortest average distance method. Lai, Liu & Hwang (1994) applied the principle of the shortest distance to propose a method by using two distance criteria (with the ideal solution and the worst solution) to evaluate multiple objectives decision problem. This principle of the shortest distance is applied with a single distance criterion, average solution, in this GDM method (The measurement criteria with ideal solution will be used as a comparison in the method ISM). The objective of ASM is to obtain the 'most average' compromise solution from the existing set of solutions provided by decision makers. Average solution represents the direction of the compromise solution. Decision makers in a group have received different solutions produced by using different MODM methods or by using the same method for an MODM application. When the set of solutions are imported into the group subsystem, an 'average value (AV)' vector called the average solution is generated. In many cases, the average solution is not an extreme point of the feasible area and is not a feasible solution for a specific MODM problem. This average solution is then compared to all solutions. The closest (shortest distance from the average solution) one is recommended as the final solution.

Let $S = (S_1, S_2, \cdots, S_n)^T$ be an n-dimensional solution vector, $S_i = (s_{i1}, s_{i2}, \cdots, s_{im})$, $i = 1,2,\cdots,n$, be a solution of an MODM application from *ith* DM member of a decision group, n be the number of decision makers (n>*2*). The MODM application consists of m objectives (m>*1*). Mathematically, the average solution method is represented as formula (8.1).

find p subject to

$$d^* = d_p = \min\{d_i; i = 1,2,\cdots,n\} = \min\{\sum_{j=1}^{m} | s'_{ij} - av_i |; i = 1,2,\cdots,n\} \qquad (8.1)$$

where

$$s'_{ij} = \begin{cases} s_{ij} / \widetilde{S}_j & if \quad \widetilde{S}_j \neq 0; \\ 0 & if \quad \widetilde{S}_j = 0, \end{cases} \quad i = 1,2,\cdots,n, j = 1,2,\cdots,m$$

$$\widetilde{S}_j = \max\{s_{ij}; i = 1,2,\cdots,n\}, \quad j = 1,2,\cdots,m,$$

$$av_j = \sum_{i=1}^{n} s'_{ij} / n, \quad j = 1,2,\cdots,m.$$

The solution process involves six major steps: (1) input of all solutions and establishment of a solution matrix S; (2) calculation of the maximum value for each decision objective and establishment of relative solution matrix S'; (3) calculation of an average solution AV; (4) estimation of distance for each objective of solutions to the average solution; (5) summary of the distances from different objectives of each solution; and (6) the solution that has the shortest distance is selected.

Step1: A solution-matrix S is built.

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1j} & \cdots & s_{1m} \\ s_{21} & s_{22} & \cdots & s_{2j} & \cdots & s_{2m} \\ & \cdots & & & \cdots & \\ s_{i1} & s_{i2} & \cdots & s_{ij} & \cdots & s_{im} \\ & \cdots & & & \cdots & \\ s_{n1} & s_{n2} & \cdots & s_{nj} & \cdots & s_{nm} \end{bmatrix}$$

Step 2. A relative solution matrix S' is established. Let

$$\widetilde{S}_j = \max\{ s_{1j}, s_{2j}, \ldots, s_{nj} \}, \qquad j = 1, \ldots, m,$$

$$S_{\max} = (\widetilde{S}_1, \widetilde{S}_2, \ldots, \widetilde{S}_m),$$

$$s'_{ij} = \begin{cases} s_{ij} / \widetilde{S}_j & if \quad \widetilde{S}_j \neq 0; \\ 0 & if \quad \widetilde{S}_j = 0, \end{cases} \quad i = 1,2,\cdots,n, j = 1,2,\cdots,m$$

We obtain

$$S' = \begin{bmatrix} s'_{11} & s'_{12} & \cdots & s'_{1j} & \cdots & s'_{1m} \\ s'_{21} & s'_{22} & \cdots & s'_{2j} & \cdots & s'_{2m} \\ & \cdots & & & \cdots & \\ s'_{i1} & s'_{i2} & \cdots & s'_{ij} & \cdots & s'_{im} \\ & \cdots & & & \cdots & \\ s'_{n1} & s'_{n2} & \cdots & s'_{nj} & \cdots & s'_{nm} \end{bmatrix}$$

Obviously,

$$s'_{ij} \in [0,1].$$

Step 3. An average solution is defined as the following:

$$AV = (av_1, av_2, .., av_m),$$

$$av_j = \sum_{i=1}^{n} s'_{ij} / n, \qquad j = 1,2,..,m.$$

Step 4. Measure the distance of each solution from the average solution. A distance matrix D for each objective of the solutions from the average solution is thus established.

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2j} & \cdots & d_{2m} \\ & \cdots & & & \cdots & \\ di1 & d_{i2} & \cdots & d_{ij} & \cdots & d_{im} \\ & \cdots & & & \cdots & \\ d_{n1} & d_{n2} & \cdots & d_{nj} & \cdots & d_{nm} \end{bmatrix}$$

where

$$d_{ij} = | s'_{ij} - av_j |, \qquad i = 1,2,.., n, \qquad j = 1,2,.., m.$$

Step 5. The distances from different objectives of each solution are summed.

$$d_i = \sum_{j=1}^{m} d_{ij}, i = 1,2,.., n.$$

Step 6. The problem now becomes to solve the following simple auxiliary problem in order to find the solution that has the shortest distance.

$$find \quad p \ subject \quad to$$

$$d^* = d_p = \min\{ d_i, i = 1,2,.., n\}, \qquad 1 \le p \le n.$$

Where d^* is the shortest total distance between the solutions and average solution, *pth* solution is the best compromise solution of this MODM problem. The value chosen for p reflects the way of achieving a compromise by minimizing the relative-distance sum of the deviations of objective from the reference point (average solution).

8.4.2 Weighting Objective Method (WOM)

This method is also called weighted shortest average distance method. The objective of this method is to combine group members' preferences and the ranking for each objective into one relative average solution. Decision makers in a group environment often receive different solutions by giving different favourite choices in using an MODM application. The aggregation of solutions is actually the aggregation of the objectives of the solutions. However, the intensity of importance of each objective is different. Thus a weight matrix for the objectives is built and weighted distance is used in the measurement distance between the solutions and average solution.

The algorithm is centred around a weight matrix W, whose elements represent the intensity of importance of objectives that decision makers prefer. Through this weight matrix an average weight vector $\overline{W}$ is obtained and is used to produce a set of weighted distances to the average solution from each solution. The 'weighted distance' to the average solution from each solution is compared and then a solution is selected which has the shortest weighted distance. This compromise solution is therefore take as the final solution if the group of decision makers is satisfied with it. Otherwise, the initial weights are modified by the decision makers and the above steps are repeated. A set of new weighted distances are then obtained and a new compromise solution is determined similarly. If this compromise solution is accepted, the process is terminated.

Let $S = (S_1, S_2, \cdots, S_n)$ be a solution vector, $S_i = (s_{i1}, s_{i2}, \cdots, s_{im}), i = 1,2,\ldots n$, be a solution of an MODM application from *ith* decision maker in a decision group, $w_{ij} \in W_j$ be the weight of *jth* objective provided by *ith* decision maker, mathematically these problems can be represented as (8.2):

$$\textit{find } p \quad \textit{subject} \quad \textit{to}$$

$$d^* = d_p = \min\{d_i, \quad i = 1,2,..,n\} = \min\{\sum_{j=1}^{m} \overline{w}_j \mid s'_{ij} - av_j \mid; i = 1,2,..,n\} \qquad (8.2)$$

where

$$s'_{ij} = \begin{cases} s_{ij} / \widetilde{S}_j, & \textit{if} \quad \widetilde{S} \neq 0; \\ 0, & \textit{if} \quad \widetilde{S} = 0, \end{cases} \quad i = 1,2,\ldots n, \quad j = 1,2,..,m$$

$$\widetilde{S}_j = \max\{s_{ij}, \quad i = 1,2,..,n\}, \quad j = 1,2,\ldots,m,$$

$$av_j = \sum_{i=1}^{n} s'_{ij} / n, \qquad j = 1,2,\ldots,m,$$

$$\overline{w}_j = \sum_{i=1}^{n} w_{ij} / n, \qquad j = 1,2,..,m.$$

The solution process involves eight steps: (1) establishment of a solution matrix *S*; (2) calculation of maximum value for each decision objective and establishment of relative solution matrix S'; (3) calculation of an average solution AV; (4) evaluation of intensity of importance for each decision objective; (5) obtaining the average weight for each objective; (6) estimation of distance for each objective of solutions to average solution; (7) calculation of the weighted distances from different objectives of each solution; and (8) the solution that has the shortest weighted distance is selected.

Step 1: To build a solution-matrix S.

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1j} & \dots & s_{1m} \\ s_{21} & s_{22} & \dots & s_{2j} & \dots & s_{2m} \\ \dots \\ s_{i1} & s_{i2} & \dots & s_{ij} & \dots & s_{im} \\ \dots \\ s_{n1} & s_{n2} & \dots & s_{nj} & \dots & s_{nm} \end{bmatrix}$$

Step 2. To establish a relative solution matrix S'. In this step, the maximum value of the elements of each column that corresponds to an objective is required. Let

$$\tilde{S}_j = \max(s_{1j}, s_{2j}, \cdots, s_{nj}), \quad j = 1, 2, \cdots, m,$$
$$S_{\max} = (\tilde{S}_1, \tilde{S}_2, \cdots, \tilde{S}_m),$$

$$s'_{ij} = \begin{cases} s_{ij} / \tilde{S}_j & if \quad \tilde{S}_j \neq 0 \\ 0 & if \quad \tilde{S}_j = 0 \end{cases} \quad i = 1, \cdots, n, \; j = 1, \cdots, m.$$

We obtain

$$S' = \begin{bmatrix} s'_{11} & s'_{12} & \dots & s'_{1j} & \dots & s'_{1m} \\ s'_{21} & s'_{22} & \dots & s'_{2j} & \dots & s'_{2m} \\ \dots \\ s'_{i1} & s'_{i2} & \dots & s'_{ij} & \dots & s'_{im} \\ \dots \\ s'_{n1} & s'_{n2} & \dots & s'_{nj} & \dots & s'_{nm} \end{bmatrix}$$

Step 3. An average solution is defined as the following:

$$AV = (av_1, av_2, \dots, av_m),$$
$$av_j = \sum_{i=1}^{n} s_{ij}, \; j = 1, 2, \dots, m.$$

Step 4. Every decision maker assigns the intensity of importance for every objective as the weight of this objective. Each weight determination involves the comparison with other elements and their relative importance to the decision maker with respect to each objective. An $n \times m$ weight matrix, W, is generated as a result of this process, where the weight is defined according to the interpretation of the scale suggested by Iz (1991) as follows (which was originally suggested by Saoty (1980)):

1-- Weak importance: experience and judgment slightly favours the objective(s) in these objectives;

3--General importance: experience and judgment strongly favour one of these objectives;

5--Demonstrated importance: an objective is strongly favoured and its dominance is demonstrated by past experience;

7-- Absolute importance: very strong evidence favouring one objective over others; and

2,4,6 intermediate values used when further compromise is needed.

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1j} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2j} & \dots & w_{2m} \\ \dots \\ w_{i1} & w_{i2} & \dots & w_{ij} & \dots & w_{im} \\ w_{n1} & w_{n2} & \dots & w_{nj} & \dots & w_{nm} \end{bmatrix}$$

$$w_{ij} \in \{1,2,3,4,5,6,7,8,9\},\ i = 1,2,..,\ n,\ j = 1,2,..,\ m.$$

Step 5. The weights at each objective from decision makers are processed to determine the average weights of these objectives. The major assumption behind the solution method is that the solutions have reflected decision makers' preferences for goals, but the decision makers in a group often have conflicting goals for each objective. The weight matrix W is expected to address the conflicts. To reveal the decision makers' preference for each objective the average weights are calculated.

$$\overline{W} = (\overline{W}_1, \overline{W}_2, .., \overline{W}_m),$$

$$\overline{W}_j = \sum_{i=1}^{n} w_{ij} / n.$$

Step 6. The distances for each objective of solutions to the average solution are first measured and a "distance matrix" D is obtained. Thus:

$$D=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2j} & \cdots & d_{2m} \\ & \cdots & & & \cdots & \\ di1 & d_{i2} & \cdots & d_{ij} & \cdots & d_{im} \\ & \cdots & & & \cdots & \\ d_{n1} & d_{n2} & \cdots & d_{nj} & \cdots & d_{nm} \end{bmatrix}$$

where

$$d_{ij} = | s'_{ij} - av_j |, \qquad i = 1,2,.., n, j = 1,2,.., m.$$

Obviously, $0 \leq d_{ij} < 1$.

Step 7. This step calculates the weighted distances d_i of each solution S_i, $i=1,2,..,n$, from distance matrix D.

$$d_i = \sum_{j=1}^{m} W_j d_{ij}, \qquad i = 1,2,..., n.$$

Step 8. The problem now becomes to solve the following simple problem:

$$\begin{aligned} &\mathit{find}\ p \quad \mathit{subject} \quad \mathit{to} \\ &d^* = d_p = \min\{ d_i, i = 1,2,.., n\}, \qquad 1 \leq p \leq n. \end{aligned}$$

The solution S_p is found as the shortest weighted distance and it is thus the best compromise solution of this MODM problem.

If the group members are not satisfied with this compromise solution, the initial set of weights are modified by each decision maker and the above steps 4 to 8 are repeated. A new compromise solution is then obtained similarly. If this solution is accepted, the process is terminated. Since the proposed decision procedure is aimed at leading towards a group compromise rather than guaranteeing one, the number of iterations to make or the time allocated to the search process can be predetermined.

8.4.3 Weighting Member Method (WMM)

The objective of this method is to combine group members' preferences and the ranking of each group member into an average solution. The degree of importance of group members is often different. Particularly, when a group meeting has a leader, this leader's preference should be reflected more in the final solution. Thus, this leader may have a higher weight for his solution than other members. In this case, the aggregation of alternative solutions is not only the aggregation of the objectives of the solutions, but also the aggregation of group members' roles.

The algorithm is centred around a weight matrix W, whose elements represent the intensity of importance of decision makers. Through this weight matrix an average weight vector $\overline{W}$ is obtained and is used to produce a set of weighted distances to the average solution for each member's solution. The 'weighted distance' to the average solution for each weighted solution is compared and a solution is selected which has the shortest weighted distance. This compromise solution is taken as the final solution if the group of decision makers are satisfied with it. Otherwise, the initial values of weightings are modified by the decision makers and the above steps are repeated. A set of new weighted distances are then obtained and a new compromise solution is determined similarly. If this compromise solution is accepted, the process is terminated.

Let $S = (S_1, S_2, .., S_n)$ be a solution vector, $S_i = (s_{i1}, s_{i2}, ..., s_{im}), i = 1,2,...n,$ be a solution of an MODM application from the *ith* decision maker in a decision group, $w_{ij} \in W_j$ be the weight of the *ith* members provided by the *jth* member. Mathematically these problems can be represented as follows:

$$\textit{find } p \quad \textit{subject} \quad \textit{to}$$

$$d^* = d_p = \min\{d_i, \quad i = 1,2,..,n\} = \min\{\overline{w_i}\sum_{j=1}^{m} | s'_{ij} - av_j |; i = 1,2,..,n\} \quad (8.3)$$

where

$$s'_{ij} = \begin{cases} s_{ij} / \widetilde{S}_j, & if \quad \widetilde{S} \neq 0; \\ 0, & if \quad \widetilde{S} = 0, \end{cases} \quad i = 1,2,...n, \quad j = 1,2,..,m$$

$$\widetilde{S}_j = \max\{s_{ij}, \quad i = 1,2,..,n\}, \quad j = 1,2,...,m,$$

$$av_j = \sum_{i=1}^{n} s'_{ij} / n, \qquad j = 1,2,...,m,$$

$$\overline{w}_i = \sum_{j=1}^{n} w_{ij} / n, \qquad i = 1,2,..,n.$$

The group aggregation solution process involves eight steps: (1) establishment of a solution matrix *S*; (2) calculation of the maximum value for each decision objective and establishment of relative solution matrix S'; (3) calculation of an average solution AV; (4) evaluation of intensity of importance for each decision maker; (5) obtain the average weight for each member; (6) estimation of distance for each objective of solutions to the average solution; (7) calculation of the weighted distances from each solution; and (8) the solution that has the shortest weighted distance is selected. These steps are similar to the weighting objective method discussed previously except steps 4 and 5.

In Step 4, every decision maker assigns the intensity of importance for every decision maker. Each weight determination involves the comparison with other decision makers' relative importance. An $n \times n$ weight matrix, W, is thus generated as a result of this process.

$$W = \begin{bmatrix} w_{11} \; w_{12} \; \cdots \; w_{1j} \cdots \; w_{1n} \\ w_{21} \; w_{22} \; \cdots \; w_{2j} \cdots \; w_{2n} \\ \cdots \\ w_{i1} \; w_{i2} \; \cdots \; w_{ij} \; \cdots \; w_{in} \\ \cdots \\ w_{n1} \; w_{n2} \; \cdots \; w_{nj} \; \cdots \; w_{nn} \end{bmatrix}$$

$$w_{ij} \in \{1,2,3,4,5,6,7,8,9\}, \; i = 1,2,.., \; n, \; j = 1,2,.., \; n.$$

In Step 5, the weights of each decision maker are processed to determine the average weights of these decision makers.

$$\overline{W} = (\overline{W}_1, \overline{W}_2, \cdots, \overline{W}_n)$$
$$\overline{W}_i = \sum_{j=1}^{n} w_{ij} / n.$$

A "distance matrix" D is obtained in Step 6.

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2j} & \cdots & d_{2m} \\ & \cdots & & & \cdots & \\ d_{i1} & d_{i2} & \cdots & d_{ij} & \cdots & d_{im} \\ & \cdots & & & \cdots & \\ d_{n1} & d_{n2} & \cdots & d_{nj} & \cdots & d_{nm} \end{bmatrix}$$

where

$$d_{ij} = | s'_{ij} - av_j |, \qquad i = 1,2,.., n, j = 1,2,.., m.$$

Step 7 calculates the weighted distances d_i of each solution S_i, $i=1,2,..,n$, from distance matrix D.

$$d_i = W_i \sum_{j=1}^{m} d_{ij}, \qquad i = 1,2,..,n.$$

The problem is then solved by the following formula:

$$\begin{aligned} &\textit{find } p \quad \textit{subject} \quad \textit{to} \\ &d^* = d_p = \min\{ d_i, i = 1,2,.., n\}, \qquad 1 \le p \le n. \end{aligned}$$

The solution S_p has the shortest weighted distance and it is thus the best compromise solution. A new solution can be generated by changing the set of weights.

8.4.4 Ideal Solution Method (ISM)

This method was used by Lai, Liu & Hwang (1994). In this method, distance from the ideal solution is used to evaluate all solutions provided by the group members. The objective of the method is to obtain a 'most optimal' compromise solution from the existing set of solutions. The most optimal compromise solution is that which is closest to the ideal solution, that is, it has the shortest distance from the ideal solution.

Let $S = (S_1, S_2,.., S_n)$ be a n-dimensional solution vector, n be number of decision makers $(n>2)$, $S_i = (s_{i1}, s_{i2},..., s_{im}), i = 1,2,...n,$ be a solution of an MODM application from ith group member, m be the number of objectives $(m>1)$. $S_0 = (s_{01}, s_{02},..., s_{0m})$ be the ideal solution. Mathematically, the ideal solution method is represented as (8.4).

find p subject to

$$d^* = d_p = \min\{d_i, i = 1,2,..,n\} = \min\{\sum_{j=1}^{m} | s'_{ij} - s_{oj} |; \quad i = 1,2,..,n\} \quad (8.4)$$

where

$$s'_{ij} = \begin{cases} s_{ij} / \widetilde{S}_j, & if \quad \widetilde{S}_j \neq 0; \\ 0 & if \quad \widetilde{S}_j = 0, \end{cases} \qquad i = 1,2,...n, \quad j = 1,2,...,m$$

$$\widetilde{S}_j = \max\{s_{ij}, \quad i = 1,2,...,n\}, j = 1,2,...,m.$$

The group aggregation process involves six steps: (1) establishment of a solution matrix *S*; (2) calculation of the maximum value for each decision objective and establishment of relative solution matrix *S'*; (3) calculation of an ideal solution; (4) estimation of the distance for each objective of solutions to the ideal solution; (5) summary of the distances from different objectives of each solution; and (6) the solution that has the shortest distance to the ideal solution is selected. These steps are similar to the average solution method except the average solution is changed to the ideal solution. The calculation of an ideal solution is the same as that discussed in Section 7.5 for the ISGP method.

When an ideal solution $S_0 = (s_{01}, s_{02}, ..., s_{0m})$ is generated, the algorithm starts to measure the distance of the ideal solution to each other solution. A distance matrix *D* for each objective of solutions to the ideal solution is thus established.

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2j} & \cdots & d_{2m} \\ & \cdots & & & \cdots & \\ d_{i1} & d_{i2} & \cdots & d_{ij} & \cdots & d_{im} \\ & \cdots & & & \cdots & \\ d_{n1} & d_{n2} & \cdots & d_{nj} & \cdots & d_{nm} \end{bmatrix}$$

where

$$d_{ij} = | s'_{ij} - s_{0j} |, \qquad i = 1,2,.., n, \qquad j = 1,2,.., m.$$

The distances from different objectives of each solution are summed:

$$d_i = \sum_{j=1}^{m} d_{ij}, i = 1,2,.., n.$$

The final solution which has the shortest distance is then found.

$$\begin{aligned} &\textit{find } p \quad \textit{subject} \quad \textit{to} \\ &d^* = d_p = \min\{ d_i, i = 1,2,.., n\}, \qquad 1 \le p \le n. \end{aligned}$$

Where d^* is the shortest total distance between the solutions and the ideal solution, *pth* solution is the best closest solution as the final compromise solution of this MODM problem.

8.4.5 Solution Analysis Method (SAM)

This method is designed to use a relaxation process for objectives based on a preliminary solution (which can be produced by ISM method). The aim of this method is to provide more interaction and negotiation for group members.

Let $S = (S_1, S_2,.., S_n)$ be an *n*-dimensional solution vector and *n* be number of decision makers (*n>2*). *Let* $S_i = (s_{i1}, s_{i2},..., s_{im}), i = 1,2,...n,$ be a solution of an MODM application from *ith* group member of a decision group, and *m* be the number of objectives (*m>1*) of an MODM application. Mathematically, the solution analysis method is represented as (8.5).

find p′ through relaxing s_p *based on*

{

find p subject to

$$d^* = d_p = \min\{d_i, i = 1,2,..,n\} = \min\{\sum_{j=1}^{m} | s'_{ij} - s_{oj} |; \quad i = 1,2,..,n\} \quad (8.5)$$

where

$$s'_{ij} = \begin{cases} s_{ij} / \widetilde{S}_j, & if \quad \widetilde{S}_j \neq 0; \\ 0 & if \quad \widetilde{S}_j = 0, \end{cases} i = 1,2,...n, \quad j = 1,2,...,m$$

$$\widetilde{S}_j = \max\{s_{ij}, \quad i = 1,2,...,n\}, j = 1,2,...,m \quad \}$$

The solution process involves two steps: (1) produce a preliminary solution such as using ISM through step 1 to 6 of ISM; (2) use the STEM method to make a relaxation in order to find a compromise solution. STEM has been discussed in Section 7.7.

.

8.5 Design and Implementation of Group Subsystem

8.5.1 Group Subsystem Work Scheme

The group subsystem is one of the three main subsystems of IMOGDSS. The subsystem is used in a group environment where decision makers' solutions need to be aggregated so as to achieve the best compromise decision.

The group subsystem includes two input schemes: electronic scheme and non-electronic scheme, and two output schemas: digital display and graphical display. Figure 8-5 shows the working scheme of the group subsystem.

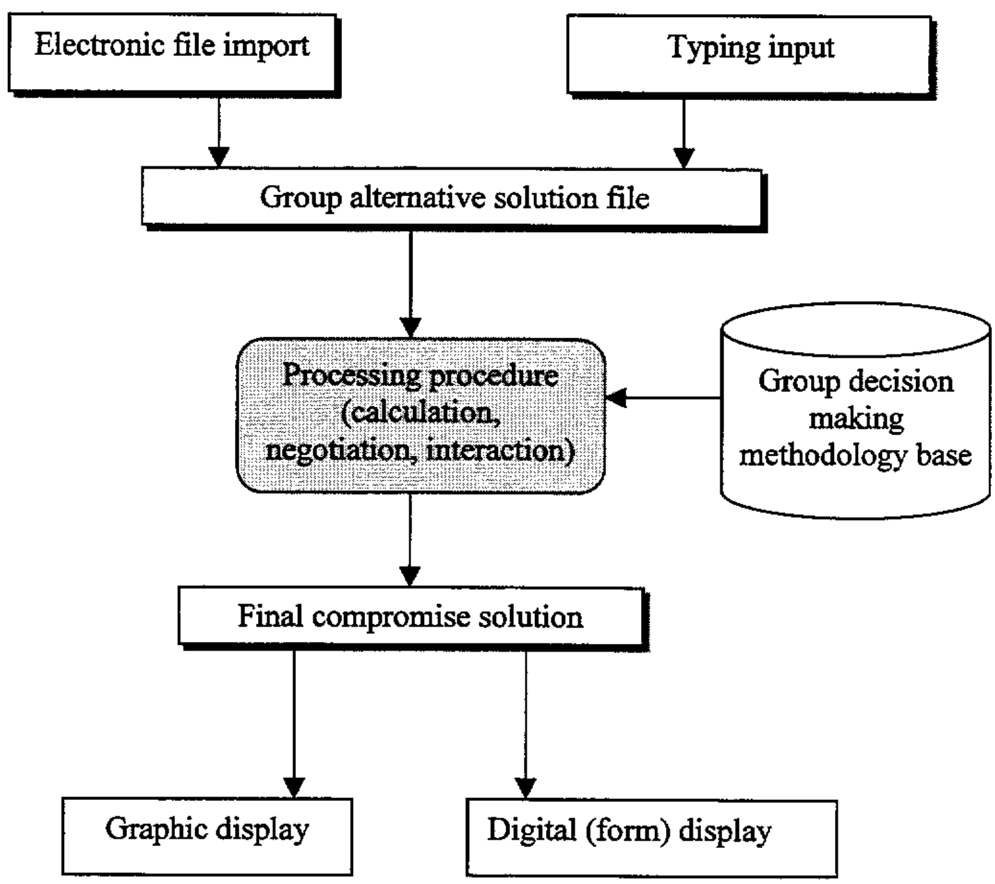


Figure 8-5 Working scheme of group subsystem

The group subsystem is activated by invoking the *group* menu from Figure 5-3. The menu produces input, import, a list of available group aggregation methods from which any method can be selected, as well as a final group solution in a digital and graphic form.

8.5.2 Individual Preferences' Representation and Input Scheme

The electronic input schema is used to read solution data from a text file. Non-electronic input (typing) scheme is used to obtain data through the keyboard. Figure 8-6 shows the first input (import) scheme, Figure 8-7 (1) and 8-7 (2) show the second input scheme. Input of objective weight is shown in Figure 8-8.

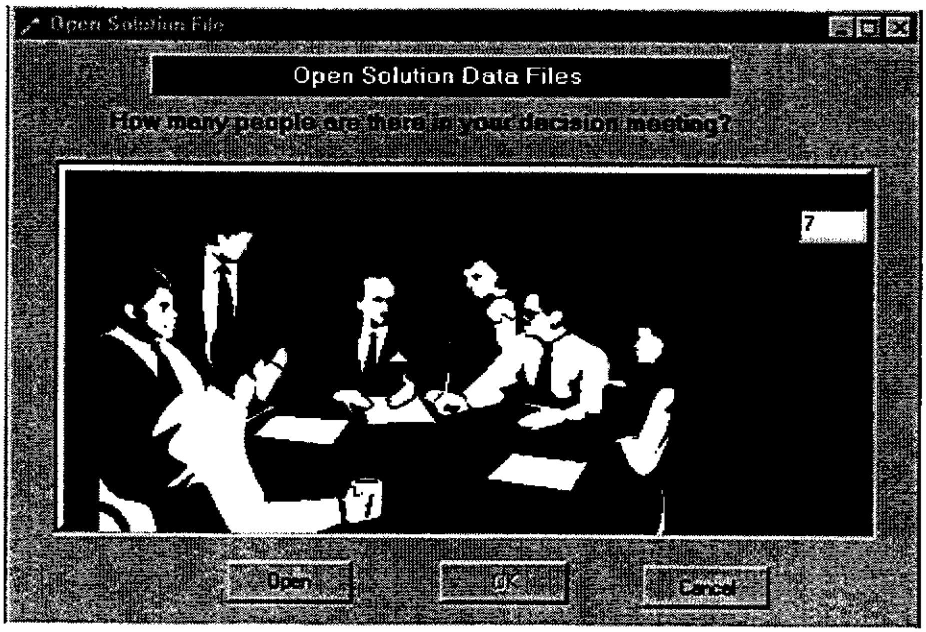


Figure 8-6 Electronic text file import—open 7 data files

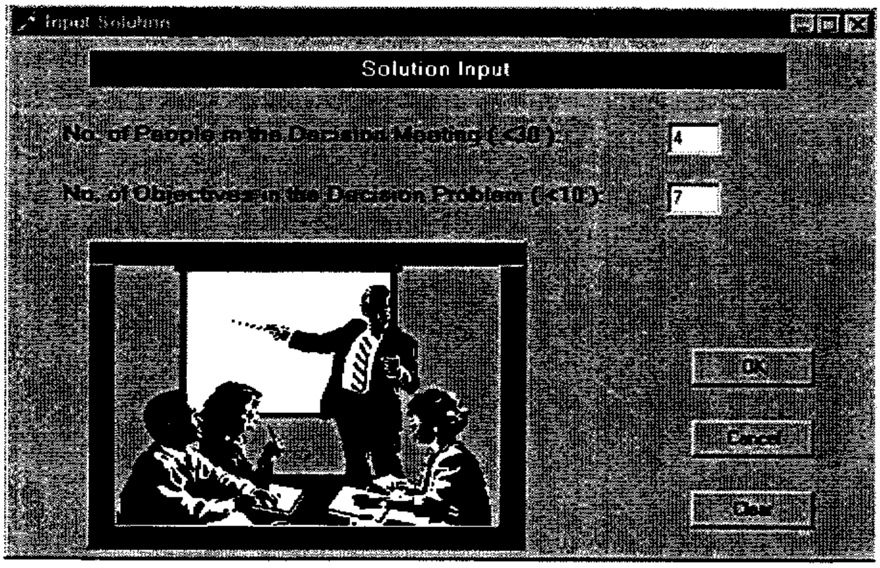


Figure 8-7(1) Group solution data input for number of members and objectives

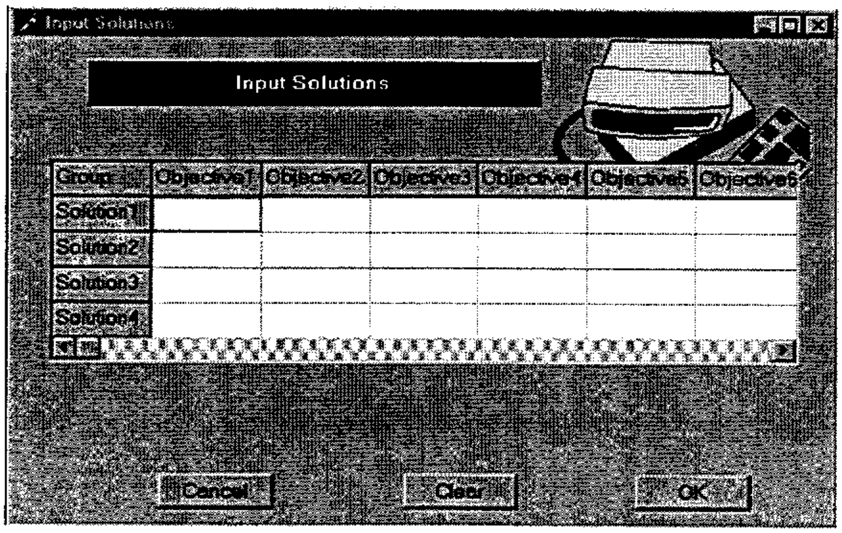


Figure 8-7(2) Group solution data input for solutions

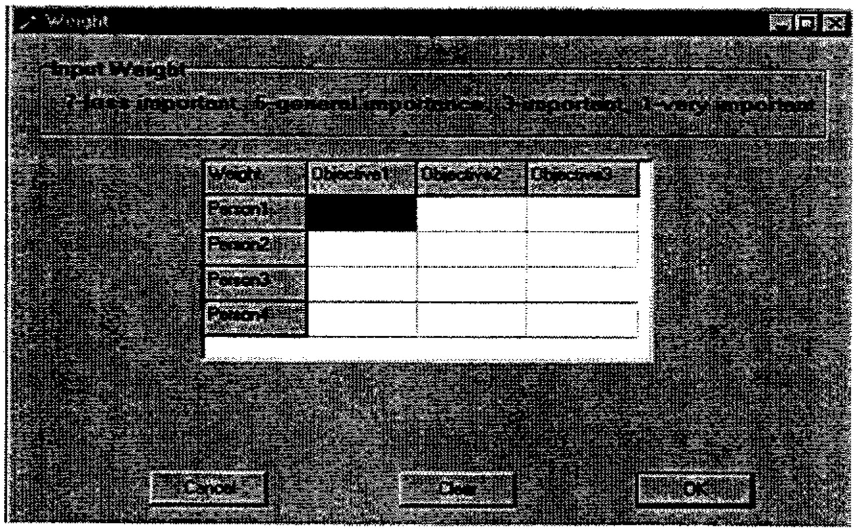


Figure 8-8 Input the weighting values

8.5.3 Performance of Group Aggregation and Reaching Consensus

All decision makers' solutions, the average solution and the ideal solution can be shown in a chart in order to view and understand the distances between average (or ideal) solution and decision makers' solutions. When the chart is shown on the screen, the users can use some control buttons to choose the type of graph, modify the properties and show the value of current items. Figure 8-9 is a graphical display and 8-10 is the corresponding digital display. Figure 8-11 is a screen of WMM in the group aggregation process. A final result is shown in Figure 8-12.

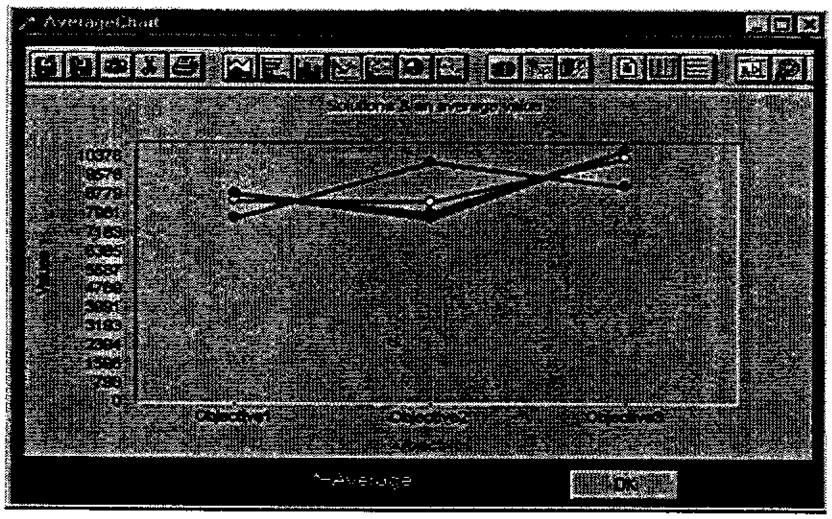


Figure 8-9 A graphical display for a group of solutions with an average solution

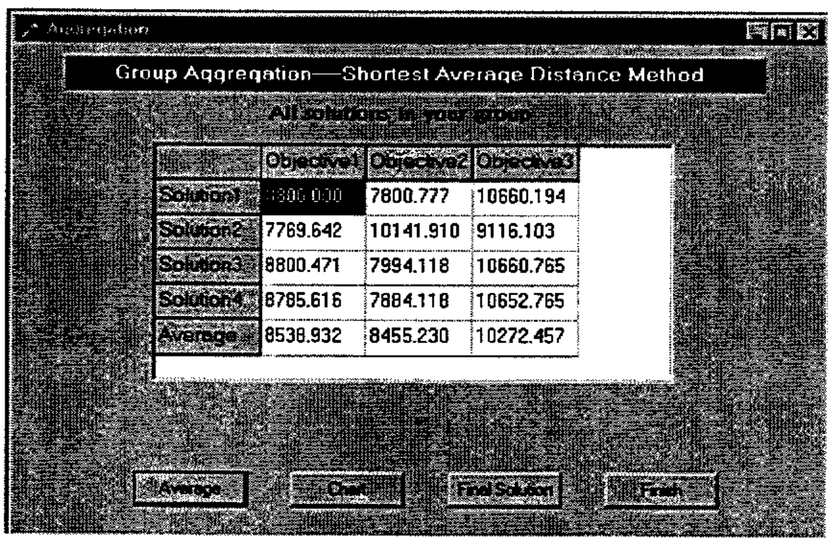


Figure 8-10 A digital display for a group of solutions with an average solution

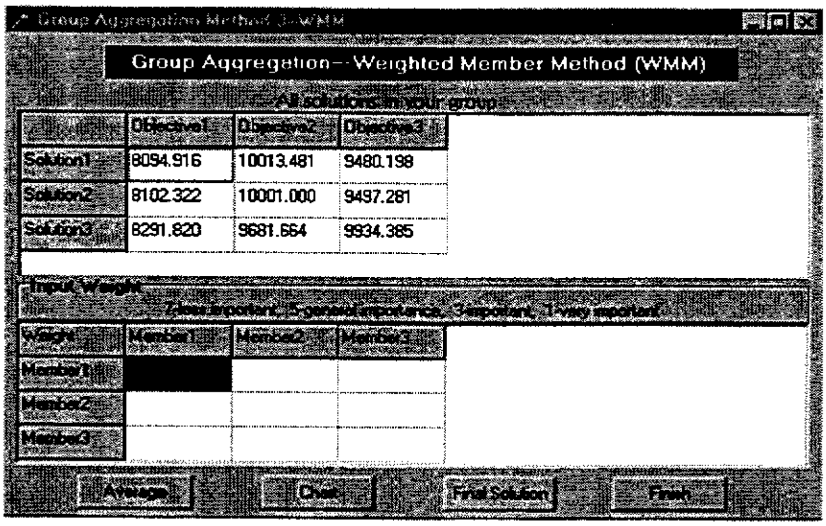


Figure 8-11 A screen of WMM in the group aggregation process

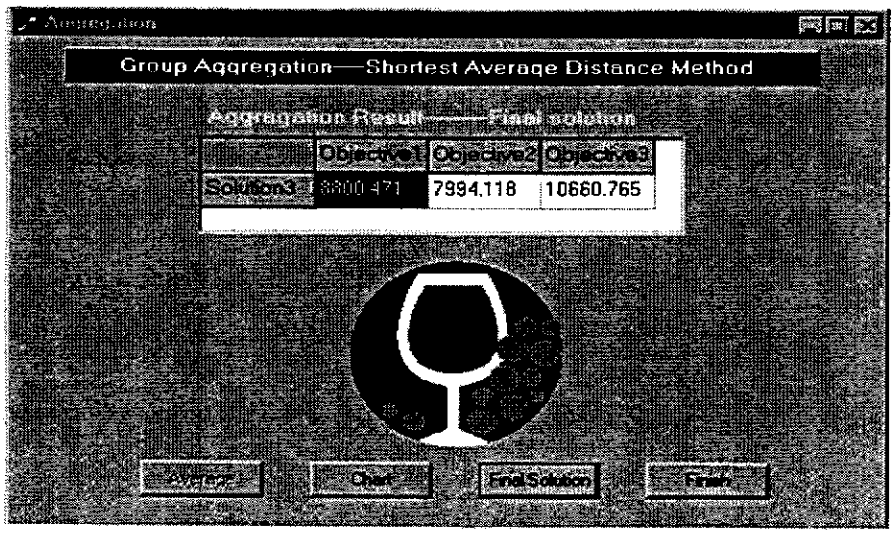


Figure 8-12 A final solution

8.6 Summary

This chapter describes the features and presents the framework of MODM-based group decision making and preference aggregation. A two-stage performance process is discussed and a group aggregation methodology base is designed. This framework has been implemented as a group subsystem imbedded in IMOGDSS. The group subsystem is developed to include a GDM methodology base involving five methods to aggregate a group of solutions that involve multiple conflicting objectives, that are provided by multiple decision makers with different business functions and preferences. This group decision making methodology base allows the decision makers to select any aggregation method in this base and allocate their MODM solutions as alternatives.

The advantages of the design include: (1) providing a GDM methodology base to use in group decision making environment; (2) supporting wide group aggregation procedures by using different GDM methods; (3) obtaining interactively the compromise solution through integrating a number of MODM solutions; (4) allowing decision makers to work at a face-to-face or dispersed environment; (5) allowing decision makers to get their solution anytime and anywhere; and (6) considering the intensity of importance for each objective (or each person) to determine the value of the relative distance.

Chapter 9

EXPERIMENTAL TESTING AND EVALUATION

9.1 Introduction

When a DSS is developed , it is the responsibility of programmers and development managers to ensure that the system will function properly. System testing, as it is called, is performed to detect errors in the logic of the system. System evaluation is performed to determine how well the system will perform and whether it meets the original specification (Senn 1990). System testing and evaluation are particularly important because it is often the last opportunity to find flaws that can cause the system to fail when it is converted to live use.

Increasingly, it is becoming recognized that experimental testing and evaluation methods should provide not only a testing or evaluation result to a system (or software tool), but also a opportunity for analysing the relationships between some variables, relevant decision makers, and the system. The study reported in this chapter has two objectives. The first one is to show the evaluation results of the functionalities of IMOGDSS from individuals and groups. The second objective is to investigate the effect of independent variables such as decision makers' education, work backgrounds etc on the dependent variables such as task performance and decision makers' confidence. For these two objectives, this chapter describes a set of experiments that examine the functionalities of the IMOGDSS and the impact of a set of independent variables on the dependent variable. Twenty-five individuals and ten groups participated in the experiment and were required to solve two completely different MODM problems. Within this experimental framework, both quantitative and qualitative results are presented and analysed. Several conclusions are drawn concerning the desirable features of IMOGDSS.

An experimental design is presented in Section 9.2. Section 9.3 provides two evaluation systems with multi-levels and multi-factors for individuals and groups. In Section 9.4, this study defines hypotheses and related experimental variables. Two applications, as decision tasks in the experiments, are described in Section 9.5. Section 9.6 shows the experimental results. Building a knowledge-based MOGDSS is an incremental process where the functionality of the system evolves as experience with its use is gained. Such system development is not a linear process. When a prototype is running, design specifications are tested, revised, and new specifications are added to accomplish needs that were not initially known. Development moves through a series of cycles before the system is finally ready for operation. This chapter only displays the end-product assessment, evaluation and analysis.

9.2 Experimental Design

9.2.1 Purposes of the Experiment

There are two aspects of concern in a system test and evaluation. Does the system work correctly and does it function well? Correctness can be tested; it is a question of right or wrong. For example, is the calculated result correct? Functionality must be evaluated; it is a question of good or bad (Klein & Methlie 1995). For instance, is the man-machine interaction well designed? Does the system fit well into the decision making process? What are the benefits of constructing a methodology base? Is the system meeting its original intended requirements and goals?

Testing the correctness of IMOGDSS, in particular, dealing with verification and validation of knowledge-based reasoning systems and mathematical models are the responsibility of developers and should be completed with the system development. The expert system in IMOGDSS is designed for well-defined problems with predefined solution sets. This makes it possible to logically verify the knowledge base and to validate the reasoning performed by the system. This chapter only concentrates on the evaluation of the functionalities of IMOGDSS from potential users' point of view. It primarily deal with issues related to users satisfactory with IMOGDSS.

Therefore, the purposes of this controlled laboratory experiment are:

- to evaluate the functionality of IMOGDSS;
- to explore the advantages of IMOGDSS by developing and testing a set of hypotheses.

9.2.2 An Experimental Design Framework

The framework of knowledge-based IMOGDSS includes ES features, MODSS features and GDSS features under the basic paradigm of decision support. ES based DSS and group based GDSS are different in several respects:

- different user relationships;
- different user expectations;
- different solution method; and
- different data processing procedures.

Therefore, when IMOGDSS is used in an individual decision maker or decision groups environment, it provides different functions. This breaks the evaluation of IMOGDSS into two different tasks.

- **Task 1: individual based experiment for evaluating IMOGDSS**

This task is taken by individual subjects. The aims of the experimental task are to evaluate the functionalities of IMOGDSS working with individual decision makers, particularly, the knowledge-based guidance and solution process of MODM methodologies. In an expert system, inference rules form chains of reasoning. Rules have been tested for correctness, completeness, and consistencies by developers. The functionalities of knowledge-based guidance must be evaluated by potential users in the experiment. The very nature of an MODSS, and the situation for which it is designed, makes it a difficult evaluation object. Since there is no predefined solution path, and the environmental context in which it is used may change from time to time, there is no single way or prescribed way of using the system (Klein & Methlie 1995). Instead, the system is a set of resources – database, knowledge base, model base and methodology base, that are placed at the disposal of the decision maker. Thus, individual decision making behaviour determines the usage of the system.

In these situations, to evaluate the IMODSS as good or bad is difficult. Therefore, an evaluation system that has multiple levels and multiple factors for IMOGDSS is needed (see Section 9.3). For exploring the benefits of MODM methodology base, the relationship between users' confidence and use of the MODM methodology base is also tested.

- **Task 2: group based experiment for evaluating IMOGDSS**

The two-stage IMOGDSS model has been discussed in Chapter 4 and Chapter 8. However, the effectiveness of such procedures needs to be tested by decision groups based on experiments. The objectives of the group based experimental study include evaluating the functionalities of IMOGDSS on the group decision support. A multiple levels and multiple factors evaluation system is used in the group experiment. A series of empirical analysis will be performed in order to explore the usefulness of GDM methodology base and the flexibility of IMOGDSS through investigating the effect of users' background and the use of the methodology base on the task performance and users confidence.

Table 9-1 describes the framework with two experimental designs. The group experimental task also employs four configurations in the group experimental design illustrated in Table 9-2. The framework structures the evaluation process, and expands the frameworks of Akoka (1981) and Iz (1992a).

Based on the experimental design framework, two sets of questionnaires were developed in order to obtain the subjects' feedback and to measure the dependent and independent variables as discussed in Section 9.4 below.

Table 9-1 Framework of two experimental design tasks

Task	Experiment Designs	Knowledge based Guide Modes	MODM Methodology Base	GDM Methodology Base
1	Individual Experiment Design	1.Novice mode 2.Intermediate mode 3.Expert mode	1.ESGP; 2.IMOLP; 3.ISGP; 4.LGP; 5.STEP; 6.STEUER; 7.ZW	
2	Group Experiment Design	1.Novice mode 2.Intermediate mode 3.Expert mode	1.ESGP; 2.IMOLP; 3.ISGP; 4.LGP; 5.STEP; 6.STEUER; 7.ZW	1.ASM; 2.WOM; 3.WMM; 4.ISM; 5.SAM

Table 9-2 Configurations of group meeting in experimental design

Configuration	Group Leader	No Group Leader
Face-to-Face	Treatment 1	Treatment 2
Computer-Mediated- Communication	Treatment 3	Treatment 4

9.2.3 Organisation of The Experiment

The experimental subjects in this study were PhD and Master level business, IS/IT, and mathematics (including operations research) students, most of them having more than 5 years work experience.

Although it may seem ideal to conduct the experimental task in this study with real managers, there were several reasons for choosing PhD and Master students as experimental subjects. As mentioned in Iz & Krajewski (1992), it is difficult to get a large group of real managers together for an extended number of meetings, including pre-study plus decision making sessions. Only three real managers participated in this experiment. Second, even if it were possible to get a large enough sample of managers, necessary for statistical power, to participate, they would have dealt with an unfamiliar case scenario just as the students. Their actual decisions would not have had the 'realistic' consequences of real life as expected from an 'ideal' experiment. Finally, the experiment was controlled for practical capabilities (education background, work background, etc.) just as it would have been controlled with real managers. While the students cannot be claimed as equivalent to practicing managers, the controlled experiment should uncover similar behavioural patterns relating to group decision making within the confines of the case scenario. Since it is necessary to replicate for statistical analysis and each company would have a different 'problem', it is impossible to run an experiment such as this in a real company.

The organisation of the laboratory experiment is described in Table 9-3 and 9-4.

Table 9-3 Individual evaluation

Educational Background \ Current Position		Real managers	Post doctorial/ research fellow	PhD students	Master/ honours student	Total
Business	Management	1		10	1	12
	Information system	1		2	3	6
Operations research/ mathematics		1	4	2		7
Total		3	4	14	4	25

Table 9-4 Experimental groups

Background / Type	Number of groups	Strong mathematics background	Strong management background	Strong IS/IT background
Face-to-face	6	2	5	4
Computer-mediated-Communication	4	4	2	4
Total	10	6	7	8

9.2.4 Experimental Procedure

The individual experiment consisted of pre-study activities, input data, selecting an MODM method(s) under knowledge-based guidance, finding a satisfactory solution using a (or more) method(s), save/output results/reports. The pre-study activities included an introduction to multiple objective linear program (MOLP), assignment of a production planning decision problem (Lai, Liu & Hwang 1994) and a food-mix selection problem (Hwang & Masud 1979) as two projects. Each subject can act as a decision maker to solve the two problems. A post-study questionnaire needs to be filled in by every subject.

The group experiment consisted of pre-study activities and group sessions. The pre-study activities included a simple lecture on MOLP, assignment of the production planning problem as an experimental project, and distribution of individual scenarios describing each subject's role in the experiment. Three subjects with similar levels of education and MOLP backgrounds were assigned to the same group. These three subjects belonged to different functional areas such as finance, marketing, and operations management. The group sessions have four different configurations, that is Treatment 1 to Treatment 4 (listed in Table 9-2). Each configuration needs to select and use one or more suitable group aggregation method(s) to reach consensus (a compromise solution). This is because some group methods only can be used in a face-to-face environment under negotiation, and some cannot be used in the decision group with a leader. During the group decision process, each member gets a satisfactory solution using the most suitable MODM method under the most fitting and proper guidance. These solutions then are aggregated into a compromise solution supported by group aggregation methodology base. This compromise solution is expected to represent the preferences of the whole group, and express the most confidence for the members of the group.

9.3 Evaluation Systems with Multi-Levels and Multi-Factors

9.3.1 An Evaluation System for IMOGDSS in Individuals

As previously discussed, there is no predefined solution path for any MODM method, and as the environmental context in which it is used may change from time to time, there is no single way or prescribed way of using the system. Therefore, a comprehensive evaluation system for IMOGDSS is necessary. The literature has been reviewed to study various evaluation variables and elements in an evaluation system (Akoka 1981; Buchanan 1994; Quaddus 1997). Based on the research methodology discussed in Chapter 3, the individual experiment employs an evaluation system that included two levels with six factors and each of them consisted of a set of sub-factors as follows:

(1) Data input

- organisation of data input
- input approaches are easy to understand
- ease of changing an input value

(2) Interactive solution process

- feedback messages are valuable during the solution process
- solution process is friendly
- solution process is easy to understand
- easy to interact with the system for finding a satisfactory solution
- satisfied with many types of solution interactive process
- make "what-if" analysis

(3) Output format

- organisation of output
- completeness of output
- output is reliable
- output is relevant

(4) Intelligent guide

- necessity of the intelligent guide
- usefulness of intelligent guide
- several guide modes suitableness
- easy to use the intelligent guide

(5) System performance

- efficiency of task performance (decision making time)
- enhancing the quality of decision making
- flexibility (this criterion distinguishes the system where the user is too constrained by the tool from those where the user has the option to explore and direct the solution process as he/she wishes)
- GUI
- effectiveness

(6) User attitude

- users' satisfaction with the ability of these MODM methods towards a satisfactory solution

- users' confidence (trust the final solution and feel confident in the final solution as being a good representative of users' preferences)

Two groups of 5-point scale (Table 9-5) have been used to assess the extent of agreement for functionality of IMOGDSS and the weighting (degree of importance) of each question with the question statements, and yes/no are used to test the correctness.

Table 9-5 Five-point scales

Point	Extent	Weighting
1	*strongly disagree*	*not important*
2	*disagree*	*less important*
3	*some what agree*	*some what important*
4	*agree*	*important*
5	*strongly agree*	*very important*

9.3.2 An Evaluation System for IMOGDSS in Groups

Group based experiments for the functionality of IMOGDSS employs an evaluation system that included two levels with seven factors and each of them consisted of a set of sub-factors as follows:

(1) Data input

- organisation of data input
- input approaches are easy to understand
- ease of changing an input value

(2) Interactive solution process

- feedback messages are valuable during solution process

- solution process is friendly
- solution process is easy to understand
- easy to interact with the system for finding a satisfactory solution
- satisfied with many types of the solution interactive process
- make "what-if" analysis

(3) Output format

- organisation of output
- completeness of output
- output is reliable
- output is relevant

(4) Intelligent guide

- necessity of the intelligent guide
- usefulness of intelligent guide
- several guide modes suitableness
- easy to use the intelligent guide

(5) Group aggregation

- aggregation methods are reliable
- enhance the quality of solution aggregation
- easy to use the group aggregation methods

- satisfied with the interactive process for finding a compromise solution
- the process for finding a compromise solution is friendly

(6) System performance

- efficiency of task performance (group aggregation time)
- enhancing the quality of group decision making
- flexibility (this criterion distinguishes the system where the user is too constrained by the tool from those where the user has the option to explore and direct the solution process as he/she wishes)
- GUI
- effectiveness

(7) User attitude

- users' satisfaction (satisfaction in the ability of these GDM methods towards a compromise solution)
- users' confidence (trust the final solution and feel confident in the final solution as being a good representative of users' preferences)

Similar to Table 9-6 shown for the individual based evaluation system, two groups of 5-point scale have been used to assess the extent of agreement and weighting for functionality of IMOGDSS in a group evaluation.

9.4 Experiment Variables and Experiment Approach

9.4.1 Hypothesis of This Experiment

The following research hypotheses are explored in the empirical study. The first of the hypotheses assesses the effect on using the MODM methodology base in an individual environment. Hypotheses 2 to 6 investigate the effect of users' background on the dependent variables.

- **Hypothesis 1**: Decision makers using more methodologies in individual decision making will have a higher confidence level in the final solution than that of decision makers using a single method (for an individual experiment);
- **Hypothesis 2**: The time it takes to reach a compromise solution is not significantly different for groups consisting of decision makers with various educational backgrounds (for a group based experiment);
- **Hypothesis 3**: The time it takes to reach a compromise solution is not significantly different for groups consisting of decision makers with various working backgrounds (for a group based experiment);
- **Hypothesis 4**: the confidence of group members in the group compromise solution is not significantly different for group consisting of decision makers with various educational backgrounds (for a group based experiment);
- **Hypothesis 5**: the confidence of group members in the group compromise solution is not significantly different for group consisting of decision makers with various working backgrounds (for a group based experiment);
- **Hypothesis 6**: Decision makers using more methodologies in a group decision making will have a higher confidence level in the final compromise solution than that of decision makers using a single method (for a group based experiment).

9.4.2 Experiment Variables

9.4.2.1 Individual Experimental Variables

Two experimental variables are defined in order to test the first hypothesis.

Independent variables:

MODM method set (methodology base): The reason for including this variable is to study the effect of the methods of MODM methodology base. Any method of the seven different MODM methods, ESGP, IMOLP, ISGP, LGP, STEP, STEUER, and ZW can be used in each experimental unit to reach a satisfactory solution. The variable was used to classify subjects into two levels:

- *using single MODM method*
- *using two or more MODM methods*

The two levels were of interest because it was employed to explore the usefulness of an MODM methodology base.

Dependent variables

User confidence: A post-study questionnaire required each subject to give the name(s) of method(s) used with respect to their confidence in the ability of the MODM method towards a satisfactory solution, trust in the final solution, and feel confident in the final solution.

9.4.2.2 Group Experimental Variables

Research in the GDSS area such as Dos Santos & Bariff (1988) and Iz & Krajewski (1992) suggests several dependent variables that can be adopted for studying the effect and impact of a group decision aid. In Iz's (1992) study, two sets of dependent measures, user satisfaction and method performance, were used to measure the efficiency and effectiveness of a group decision procedure. The current study employed two sets of dependent variables, user confidence and task performance to measure the effect of educational background, working background and methodology base on the dependent measures.

Independent variables:

Educational background: Through this variable the subjects were classified as one of three possible levels, business, information systems and maths. These levels were of interest because each represented subjects from different background groups. While business students were mostly juniors and had stronger knowledge in business management, information systems students had wide range of knowledge in system development, and maths people have strong knowledge of MOLP.

- *business / management*
- *information systems / information technology*
- *linear programming / maths background*

Working background (work experience): this variable was classified as the three possible levels:

- *business / management*
- *information systems / information technology*
- *linear programming / maths background*

GDM method set (Methodology base): the reason for including this variable is to compare the use of one method with more methods in leading towards a group compromise solution. Different subsets of five different GDM methods, ASM, WOM, WMM, ISM and SAM, can be used in each experimental unit. The variable is used to classify subjects (groups) into two levels:

- *single GDM method*
- *two or more GDM methods*

The two levels were of interest because it was employed to explore the usefulness of a GDM methodology base.

Dependent variables

Task performance (decision making **time**)

User confidence: trust in the final compromise solution and feel confident in the group consensus solution as being a good representative of users' preferences

The two dependent variables, *task performance (time)* and *user confidence*, were used to measure the GDM methodology base' performance. The subjects in this study were well aware that the determining factors of their grades in the laboratory study would not necessarily be the length of discussion time or the number of methods but the content of discussions and the rationale behind subjects' arguments.

9.4.3 Experiment Approach

Only FTF based (a decision meeting in a decision room) group laboratory experimental evaluation methodology is described in this section. CMC based (Treatment 3 to 4) and individual experimental procedure is similar and easier than this procedure. The procedure is as follows:

1. Group forming and group members

The experimental subjects in this study are Masters or PhD level business/ IS / Maths students. Subjects participated in the laboratory experiment have basic background of management, information system and mathematics. Few real managers and mathematician participated in this laboratory experiment.

2. Pre-study

All members participated a pre-study decision making session to help them to understand the objective of this experiment. The pre-study activities included a seminar on intelligent multiple objectives group decision making system, a simple lecture on MOLP, a production planning problem as a decision application project. However some participants have a high level of MOLP, linear programming, DSS, product plan, ES or GDSS background. The pre-study activities also included the distribution of individual scenarios describing each subject's role in the experiment. The final pre-study event was a meeting with each group during which subjects were asked to decide the managerial role they would like to play during the experiment. They would have to be familiar with the case scenario of an application.

3. Controlled experiment

The experiment is controlled for practical capabilities. The controlled experiment should uncover similar behavioural patterns relating to group decision making within the confines of the case scenario, since it is necessary to replicate for statistical analysis.

4. Group decision task

The decision task involved a project that required the groups to find a compromise solution to an aggregate planning model with three conflicting objectives. The conflict arises because an improvement in one objective can only be made by trading off one or more of the rest of the objectives on the efficient frontier. Each subject in a group was responsible for one of the three functional areas of the company. Preceding the experiment, subjects were provided with individual scenarios that described their roles and provided historical information about their particular area of responsibility in the company.

5. Laboratory experimental environment requirement

- Each group has a session for an application through a FTF configuration or two configurations (FTF and CMC);
- The FTF group sessions are held in an office equipped with a computer terminal and a printer, called a 'decision room';
- Groups are allowed to be of three to four members;

- Groups are allowed a maximum of one hour to find a compromise solution to an aggregate decision problem;

- A technical assistant, called the facilitator, present during the group sessions, kept track of the time, recorded the sessions on tape for future reference, and helped to operate the IMOGDSS in case of unexpected technical difficulties;

6. Laboratory experimental stages

- The facilitator shows the application.

- Each member assumes a role in the company (thinking about the goals, weights of these conflicting objectives).

- Each participant first uses the intelligent subsystem to get a recommendation about which MODM method is most suitable.

- They then use the MODM method to obtain a satisfactory solution.

- These solutions are saved and passed to the facilitator.

- The facilitator opens these solution files in the group subsystem and shows them to all members.

- A group aggregation method is used and a compromise solution is obtained. If all members are satisfied with the solution, then this solution is the final solution. The decision meeting breaks up.

- Another group aggregation method is used if the compromise solution from the first method is not accepted by the group. The members may be needed to assign their weights for each objective of the decision problem, or each other member. A compromise solution is then formed.

- If the group members are not satisfied with this compromise solution, the initial set of weights are modified by each of the members and the above step is repeated, or a relaxation is used. This process depends on which group aggregation method is selected from the GDM methodology base. A new compromise solution is then obtained similarly. If this solution is accepted, the process is terminated.

- Otherwise, a new group aggregation method is used. The steps mentioned above is repeated. Since the proposed decision procedure is aimed at leading towards a group compromise rather than guaranteeing one, the number of iterations to make or the time to allocate to the search process can be predetermined.

- Each participant needs to fill in a group decision-maker evaluation form after the last compromise solution is found and the session is ended. They are required to fill in the questionnaire regarding their experiences.

As IMOGDSS was designed to provide a more active group decision support for multiple objective decision making, we have organised a small group of subjects to conduct several experimental CMC decision making processes. A typical experiment includes a number of subjects at different geographical locations. For example, some of them are in their offices and others are at home. We have also arranged for some subjects to try IMOGDSS during the day time, while others try it in the evening.

9.5 Decision Task

Two different decision tasks, a production planning decision (Lai 1995), and a food-mix selection problem (Hwang & Masud 1979), are used in individual experiment. They have different number of objectives, constraints and variables. These decision making problems are solved by different subjects, using different MODM methods, under different intelligent guide modes. The production planning decision task (Lai 1995) is also used in the group experiment.

This manufacturing company has six machine type - milling machine, lathe, grinder, jig saw, drill press and band saw - whose capacities are to be devoted to production of three products X1, X2 and X3. Among various possible objectives in an aggregate production plan, only three were explicitly considered in this application. The first objective is to maximise net profits, which was defined as the difference between revenues, and the cost of regular and overtime labour, plus the cost of hiring and firing, inventory cost, and other production costs besides labour. The second objective of the aggregate production planning application is to maximise quality of products. The third objective in the model is to maximise worker satisfaction.

The MOLP problem is shown in Table 9-6 and presented in model (9.1). The three objectives in the model are represented as 1, 2 and 3. There are eight decision variables, and eight technical constraints. The relative importance of an objective is assumed to vary among individual subjects. Hence, despite an agreement among the subjects on the set of objectives for the group decision problem, the priority ranking of the objectives are assumed to be different for different members of the group. Due to the conflict among objectives, it is impossible to find a solution to this problem that optimises all three objectives simultaneously. However, it is possible to determine an efficient solution to the problem. A solution to a multiple objective problem is efficient only if it corresponds to a non-dominated criterion vector. That is, a solution point in the decision space is efficient if it is not feasible to move from it and increase an objective value without decreasing the value of at least one other objective.

Table 9-6 Production planning problem

Machine	Product X1	Product X2	Product X3	Machine hours available	Unit cost ($)
Milling machine	12	17	0	1400	20
Lathe	3	9	8	1000	30
Grinder,	10	13	15	1750	25
Jig saw	6	0	16	1325	25
Drill press	0	12	7	900	35
Band saw	9.5	9.5	4	1075	20
Objectives:					
Profits	50	100	17.5		
Quality	92	75	50		
Worker satisfaction	25	100	75		

A group of subjects are assumed to be responsible for determining a compromise solution to the planning problem. Each subject in a group is responsible for one of the three functional areas of the company. Prior to the experiment, subjects are provided with individual scenarios that described their roles and provided historical information about their particular area of responsibility in the company.

$$
\left.\begin{array}{ll}
Max & f_1(x) = 50x_1 + 100x_2 + 17.5x_3 \\
Max & f_2(x) = 92x_1 + 75x_2 + 50x_3 \\
Max & f_3(x) = 25x_1 + 100x_2 + 75x_3 \\
\multicolumn{2}{l}{subject\ to:} \\
& g_1(x) = 12x_1 + 17x_2 \le 1400 \\
& g_2(x) = 3x_1 + 9x_2 + 8x_3 \le 1000 \\
& g_3(x) = 10x_1 + 13x_2 + 15x_3 \le 1750 \\
& g_4(x) = 6x_1 + 16x_3 \le 1325 \\
& g_5(x) = 12x_1 + 7x_3 \le 900 \\
& g_6(x) = 9.5x_1 + 9.5x_2 + 4x_3 \le 100 \\
& \\
& x_1, x_2, x_3 \ge 0
\end{array}\right\} \quad (9.1)
$$

Another application used in this experiment is a food-mix selection problem (Hwang & Masud 1979) called a Nutrition problem.

The nutrition problem is to find the quantities of certain foods that should be eaten to meet certain nutritional requirements at a minimum cost. Information on six foods is given in Table 9-7. The problem is to find the quantities of these foods that satisfy:

(1) the minimum daily requirement of vitamin A and iron; and the balanced daily requirement of food energy and protein;

(2) the minimum cholesterol daily intake;

(3) the minimum carbohydrate daily intake; and

(4) the minimum cost.

Table 9-7 Nutrition problem

	Milk	Beef	Eggs	Bread	Lettuce	Orange	Recommended Daily allowance
Vitamin A	720	107	7080	0	134	1000	5000
Food energy	344	460	1040	75	17.4	240	2500
Cholesterol	10	20	120	0	0	0	
Protein	18	151	78	2.5	0.2	4	63
Carbohydrate	24	27	0	15	1.1	52	
Iron	0.2	10.1	13.2	0.75	0.15	1.2	12.5
Cost	0.225	2.2	0.8	0.1	0.05	0.26	

The maximum daily intakes of the food are: 6 pints of milk, 1 pound of beef, ¼ dozen eggs, 10 ounces of bread, 10 ounces of lettuce and salad, and pints of orange juice.

Let the daily diet requirements of milk, beef, eggs, bread, lettuce and salad, and orange juice be X1, X2, X3, X4, X5 and X6, respectively. Then, the MODM problem is described in model (9.2).

$$
\left.
\begin{array}{ll}
Max & f_1(x) = -0.225x_1 - 2.2x_2 - 0.8x_3 - 0.1x_4 - 0.05x_5 - 0.26x_6 \\
Max & f_2(x) = -10x_1 - 20x_2 - 120x_3 \\
Max & f_3(x) = -24x_1 - 27x_2 - 15x_4 - 1.1x_5 - 52x_6 \\
\multicolumn{2}{l}{subject\ to:} \\
& g_1(x) = 720x_1 + 107x_2 + 7080x_3 + 134x_5 + 1000x_6 \geq 5000 \\
& g_2(x) = 0.2x_1 + 10.1x_2 + 13.2x_3 + 0.75x_4 + 0.15x_5 + 1.2x_6 \geq 12.5 \\
& g_3(x) = 344x_1 + 460x_2 + 1040x_3 + 75x_4 + 17.4x_5 + 240x_6 \geq 2500 \\
& g_4(x) = 18x_1 + 151x_2 + 78x_3 + 2.5x_4 + 0.2x_5 + 4x_6 \geq 63 \\
& g_5(x) = x_1 \leq 6 \\
& g_6(x) = x_2 \leq 1 \\
& \\
& x_1, x_2, x_3, x_4, x_5 . x_6 \geq 0
\end{array}
\right\} \quad (9.2)
$$

9.6 Analysis of the Experiment Results

9.6.1 Results for IMOGDSS in Individual Users

In this post experimental questionnaire, each individual subject answered their education background, work background and assigned extents of

1- strongly disagree

2-disagree

3-some what agree

4-agree

5-strongly agree

to the all questions (factors) of the Evaluation Form (Appendix 2). The questionnaire data is processed through related descriptive statistics using SPSS. The higher mean (close to 5) indicates that this factor is very satisfactory to users in its functionality. The results are displayed in Table 9-8. The last column of Table 9-8, standard deviation, shows that there are lower error estimate and higher reliability in the group of data.

9.6.2 Results for IMOGDSS in Groups

The evaluation results of IMOGDSS for 7 main factors and 28 sub-factors in the groups are displayed in Table 9-9. Similar to individuals, the extent of factors is defined into 1 to 5. It is found that usefulness of intelligent guide, enhance the quality of GDM, ease of use of group aggregation methods and satisfied with the interaction process obtained very high scores. This shows that users are satisfied with these functions provided by IMOGDSS. Almost all factors' scores are higher than 4. The result represents that IMOGDSS is perceived to be successful in its functionality. The values of standard deviation also indicate the error is in an acceptable quality limit.

Table 9-8 Descriptive statistics (individuals)

Factors	**Sub-factor**	**Mean Extent (ξ)**	**STDEV**
Data input	organisation of input	4.64	0.56862
	input approach is easy to understand	4.52	0.65319
	ease to change input value	4.44	0.50662
Interactive process	feedback messages valuable	4.24	0.52281
	the solution process is friendly	4.64	0.48989
	easy to interact with the system	4.56	0.58309
	solution process is easy to understand	4.76	0.43588
	make "what-if" analysis	4.72	0.45825
Output format	organisation of output	4.68	0.55677
	completeness of output	4.72	0.45825
	output is reliable	4.72	0.54160
	output is relevant	4.68	0.47609
Intelligent guide	necessity of intelligent guide	4.68	0.47609
	usefulness of intelligent guide	4.68	0.47609
	guide mode suitableness	4.64	0.48989
	easy to use the intelligent guide	4.72	0.45825
Performance	efficiency of task performance	4.48	0.50990
	enhance the decision making quality	4.44	0.50662
	flexibility	4.56	0.50662
	GUI	4.64	0.48989
	effectiveness	4.6	0.5
User attitude	user satisfaction	4.6	0.5
	user confidence	4.68	0.47609

Table 9-9 Descriptive statistics (groups)

Factors	Sub-factor	Mean Extent (ξ)	STDEV
Data input	organisation of input	4.43333	0.67891
	input approaches are easy to understand	4.5	0.62972
	ease to change input value	4.4	0.49827
Interactive process	feedback messages valuable	4.26666	0.52083
	the solution process is friendly	4.56666	0.56832
	easy to interact with the system	4.5	0.62972
	solution process is easy to understand	4.63333	0.49013
	make "what-if" analysis	4.6	0.49827
Output format	organisation of output	4.6	0.62145
	completeness of output	4.73333	0.44977
	output is reliable	4.63333	0.55605
	output is relevant	4.66666	0.47946
Intelligent guide	necessity of intelligent guide	4.6	0.56324
	usefulness of intelligent guide	4.73333	0.44977
	guide mode suitableness	4.63333	0.49013
	easy to use the intelligent guide	4.7	0.46609
Group aggregation	aggregation methods are reliable	4.52631	0.51298
	enhance the quality of GDM	4.84210	0.37463
	easy to use group aggregation methods	4.78947	0.41885
	satisfied with the interactive process	4.89473	0.31530
	aggregation process is friendly	4.73684	0.45241
Performance	efficiency of task performance	4.43333	0.50400
	enhance the quality of decision making	4.53333	0.50741
	flexibility	4.58620	0.50123
	GUI	4.5	0.50854
	effectiveness	4.56666	0.50400
User attitude	user satisfaction	4.63333	0.49013
	user confidence	4.66666	0.47946

Based on the evaluation results and discussion with the subjects afterwards, this research has summarised the following **comments:**

- An MODM methodology base is very useful for users to use any suitable method or several methods to solve their specific application, to improve their satisfactory degree to get satisfied solutions, and to increase their confidence;
- A knowledge base is very necessary and helpful to be able to guide users to select the most suitable MODM method;
- A GDM methodology base facility helps decision groups to resolve different aggregation problem more easily and make group members satisfactory with the compromise solution and confident with their preferences;
- The IMOGDSS is very flexible. Users can join a group decision making process any time, at any location and on any type of computer;
- The IMOGDSS is very user friendly with a high degree of interaction. The GUI-based interface is very helpful for decision makers to make decisions;
- The IMOGDSS has powerful functionality to match the requirement of decision maker for solving multiple objective decision problems.

9.6.3 Analysis of Variance and Tests of Hypotheses

9.6.3.1 Analysis of variance in individuals

A one factor fixed effects analysis of variance model (Hughes & Grawoig 1971; Everitt 1998) was used to determine the effect of the independent variables on the dependent measures in individuals. The form of the full model is as follows (9.3):

$$Y_i = \mu.. + \alpha_i + \varepsilon_i \qquad (9.3)$$

where, μ is overall mean, ε_i is error of dependent variable Y_i, and α represents the effect on dependent variable Y of the type of individual decision procedure using one or more methodologies in this study. The general form of the model (9.3) therefore can be written as follows (9.4):

$$[confidence] = \mu.. + \alpha[methodology\ base] \qquad (9.4)$$

Table 9-10 and 9-11 present the analysis of variance results.

Table 9-10 ANOVA results (Tests of between-Subjects effects)

Effect	**Dependent Variable**	**F**	**Significance**
Methodology base	**Confidence**	5.792	0.025

Table 9-11 Cell means for the effects

Independent Variables	**Dependent Variables: Confidence**					
	Level	**Mean**	**N**	**Std. Deviation**	**T-Value**	**P-Value**
Methodology base	**One**	4.3750	8	.5175	-2.407	0.0125
	More	4.8235	17	.3930		
	Total	4.68	25	.4761		

Table 9-10 shows that the decision makers' confidence in the final solution based on the number of MODM methods he/she used. The confidence of decision makers using two or more MODM methods was significantly higher than that of decision makers using only one method. "Hypothesis 1" claimed that decision makers using the multiple MODM methodologies would have high confidence in the final solution. The results in Table 9-10 support this hypothesis. Table 9-11 displays the mean and standard deviations of the two levels. P-value (0.0125) indicates the two groups are obviously different. The results tell that the MODM methodology base is very helpful for users to obtain a satisfactory solution and enhance the confidence of users for the final solution.

9.6.3.2 Analysis of variance in groups

A three factorial fixed effects analysis of variance model (Hughes & Grawoig 1971; Everitt 1998) was used to determine the effect of the independent variables on the dependent measures. The form of the full model is as follows (9.5):

$$Y_{ijk(n)} = \mu.. + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk} + (\alpha\beta\gamma)_{ijk} + \varepsilon_{ijk} \quad (9.5)$$

Where, μ is overall mean, ε_{ijk} is error of dependent variable Y_{ijk}, α, β and γ represent respectively the effect on dependent variable Y of the type of group decision procedure used and subjects' level of education background, work background and the use of methodology base. After testing the usual assumptions of the fixed effects model and performing indirect tests for model adequacy, the interaction term $\alpha\beta$, $\alpha\gamma$, $\beta\gamma$ and $\alpha\beta\gamma$ on Y was found statistically insignificant and therefore, dropped from further analysis.

The general form of the model (9.5) that was found satisfactory for the analysis of results discussed in this study about group decision is as follows (9.6 and 9.7):

n=1:

$$\begin{bmatrix} task\ performance\ (time) \\ \end{bmatrix} = \mu .. + \alpha \begin{bmatrix} education \\ backgroung \end{bmatrix} + \beta \begin{bmatrix} work \\ background \end{bmatrix} \qquad (9.6)$$

n=2:

$$\begin{bmatrix} \\ confidence \end{bmatrix} = \mu .. + \alpha \begin{bmatrix} education \\ backgroung \end{bmatrix} + \beta \begin{bmatrix} work \\ background \end{bmatrix} + \gamma \begin{bmatrix} methodology \\ base \end{bmatrix} \qquad (9.7)$$

Table 9-12, 9-13 and 9-14 summarise the analysis of variance results for groups.

Table 9-12 ANOVA results (Tests of Between-Subjects Effects)

Effect	Dependent Variable	F	Significance
Methodology base	**Task performance (Time)**	*	*
	Confidence	18.262	.001
Education background	**Time**	.000	1.000
	Confidence	.253	.620
Work background	**Time**	3.665	.068
	Confidence	1.264	.273

model: intercept + METHOD +EDUCATION+WORK +METHOD*EDUCATION +METHOD*WORK +EDUCATION*WORK+METHOD*EDUCATION*WORK

Table 9-13 Descriptive statistics for cell mean of main effects

Independent Variables	**Dependent Variables**				
	Level	**Task performance (Time)**		**Confidence**	
		Mean	**Std. Deviation**	**Mean**	**Std. Deviation**
Methodology base	**One** (*single GDM method)*	**	**	4.2	.4216
	More (*two or more GDM methods)*	**	**	4.9	.3078
Education background	*business / management*	24	4.1404	4.6667	.4880
	information systems /technology	26.6667	5.0000	4.6667	.5000
	linear programming / maths background	25	5.4772	4.6667	.5164
Work background	*business / management*	21.8750	4.0311	4.6250	.5
	information systems / technology	27.5	4.6291	4.75	.4629
	linear programming / maths background	25.000	5.4772	4.6667	.5164

Table 9-14 Descriptive statistics for cell mean of main effects

Independent Variables	Dependent Variables		
	Level	**Task performance (Time)**	**Confidence**
		Significant	**Significant**
Methodology base	**One** (*single GDM method)*	**	0.000
	More (*two or more GDM methods)*		
Education background	*business / management*	0.68	1.000
	information systems /technology		
	linear programming / maths background		
Work background	*business / management*	0.023	0.844
	information systems / technology		
	linear programming / maths background		

** not tested

The results in Table 9-12 show the time is insignificantly correlated with the education background (Sig.=1.00). The results support "Hypothesis 2" that posited no significant difference between the time spent by groups consisting of decision makers with different educational background in finding a compromise solution and the time spent by their counterparts. But the results show the time is significantly dependent on the work background (Sig.=0.68<1). The results are contrary with "Hypothesis 3". It is observed from Table 9-12 that the dependent variable, confidence, is insignificantly correlated with the independent variable education background and work background. The results support "Hypotheses 4 and 5" that addressed the decision maker's confidence in the final compromise solution based on the level of his/her education background and work background. The results in Table 9-12 also indicate significant difference in subject confidence with respect to the measures for using multiple methods provided in the methodology base. This supports "Hypothesis 6".

Table 9-13 displays the means and standard deviation of two dependent variables on each level of independent variables. Table 9-14 analyses significance between these levels. The results further indicate that the two groups testing data related to the number of methodologies used are obviously different. It has been found that the level of methodology base had a significant effect on the confidence measure (Sig.=0.001). Groups using more methodologies had high confidence in their compromise solutions. That is, **confidence of decision makers using the multiple methods was significantly higher than that of decision maker using only one method.** This finding supports strongly to what is claimed in "Hypothesis 6". This finding may be partially explained by the fact that decision makers explored multiple ways to aggregate their solutions and used some methods to confirm the aggregated results so that they felt very confident about the final solution.

Table 9-14 shows that three levels of education backgrounds were not obviously different to task performance measure and here is no significant difference on the confidence measure. But the three levels of work background had a non-significant effect on the confidence and had a significant difference (0.023) on the task performance (time). Based on Tables 9-13 and 9-14, it may be explained that subjects who have business/management work background are familiar with the application problem in this experiment, and thus they can reduce the time of arriving at the aggregated solution.

9.7 Summary

This chapter illustrates the ways in which the system evaluation and experimental research methodology described in Chapter 3 can be used in IMOGDSS. This chapter presents the evaluation results for IMOGDSS from experiments done with individuals and groups. The results show that IMOGDSS can be used effectively to support multiple objective decision making for single decision makers and decision groups. It has the ability to support decision makers to select the most suitable MODM method under a knowledge-based guidance. It is suitable for a wide range of group decision environment. The effectiveness of two-stage IMOGDSS model has been tested successfully by decision groups-based experiment. It is observed that the 'user confidence' is significantly correlated with the use of 'methodology base' of IMOGDSS. There are two main conclusions to be drawn from this study.

- The conceptual framework of integrating MODSS, ES and GDSS to deal with MODM problem in individual/group decision making under a knowledge-based intelligent architecture is quite a success and has a series of advantages.

- The framework has been implemented effectively as a new configuration of group decision support software-IMOGDSS prototype, which provides GUI-based hierarchical procedure for solving MODM problems with intelligent guidance in a group. The prototype posses powerful functions and higher user satisfaction.

Chapter 10

CONCLUSIONS AND FUTURE RESEARCH

10.1 Introduction

This chapter first identifies the framework and prototype of IMOGDSS and its applications in Section 10.2. Next, Section 10.3 recapitulates the entire research and highlights the main findings from previous chapters. The future research is then provided in Section 10.4.

10.2 Summary of This Study

The objectives of this research were (i) to develop a conceptual framework and its prototype to extend the application capability of a category of MODSS techniques through providing a complete method management function, (ii) to explore the combined functionalities of ES and MODSS through embedding an intelligent front-end, and (iii) to develop a new configuration of dealing with MODM models in a GDSS framework. Ultimately, an integrated system of MODSS, IDSS and GDSS was generated, which contains a sufficient number of MODM methods, and provides the guide to select the most suitable method, and also aggregates individual decision makers' preference to produce a compromise solution supported by a GDM methodology base. This integrated system is called IMOGDSS, which was also evaluated in a laboratory experimental setup.

This thesis has identified various issues of MODSS, GDSS, knowledge base, and system co-ordination that are central to such integration of vastly different technologies. This thesis has expanded the scope and extent of the applications of MODSS to meet these requirements for MODM methodology base, method selection guide, and individual/group decision making. This thesis proved that it is possible to construct a functional integration at the application system level of MODSS, GDSS and IDSS techniques. It has also indicated, what is the structure of this integrated system and its components, and how to do balance between MODM, DSS, intelligent technology and GDM in the effort to best exploit the opportunities of the integrated system. Based on the research all the research questions provided in Chapter 1 have been addressed comprehensively.

10.3 Major Conclusions

10.3.1 Findings

This thesis has proposed an integrated theoretical framework of knowledge based IMOGDSS that takes advantage of the intelligence, flexibility, usefulness, effectiveness and active aspects of multiple technologies. In order to demonstrate the technical feasibility of the proposed IMOGDSS architecture, a prototype implementation of the IMOGDSS framework was down using DELPHI and CLIPS based GUI techniques. The major findings providing answers to the research questions and results from the prototype testing and evaluation are outlined below:

- IMOGDSS resulted in a successful integration of KB technologies into an MOGDSS framework by means of applying a knowledge base. A knowledge based guide facilitates the interaction between the decision maker and IMOGDSS and guides the users to select and use the most suitable MODM method for their specific application.

- The functional and resource integration framework of IMOGDSS, where MODSS, IDSS and GDSS are integrated, are effective to reduce communication barriers, enhance group capabilities, improve the efficiency of MODM problem solving, increase the depth of decision analysis, and lead to greater satisfaction in the multiple objectives group decision making process.

- IMOGDSS allows the decision makers to participate in the decision making process with an opportunity of learning how the individual components of the system interact and how these interactions affect the performance of the system. The use of a methodology base enhances the confidence of the decision makers in the final solution of their MODM problems.

- The collaboration between MODSS and GDSS, particularly, the framework of two-stage group decision making procedure are able to help the decision makers to analyse, understand and interact co-operatively in the group decision making process and to overcome information overload.

- The group aggregation methodology base provides an opportunity to represent and aggregate the individual members' preferences, and provides an environment to compare and refine the compromise solution in order to increase the confidence of each group member on the compromise solution.

- The laboratory based experiments with IMOGDSS has shown extremely encouraging outcomes.

10.3.2 Contributions

The research questions pursued in this study are new, creative and important in MCDSS and GDSS fields. The research is quite complex, as it involves the integration of several kinds of information systems which combines the strong competence of each system and complements the abilities among them. This research deals with both theoretical framework development and software implementation. It bridges the gaps between previous and this study. **Five major contributions** of the research are listed below:

- This study shows the conceptual framework of integrating MODSS, ES and GDSS effectively to deal with MODM problem in individual/group decision making under a knowledge-based intelligent architecture. Three dimensions, MODSS, ES, GDSS are combined to overcome the limitations of each and maximally enhance the competence of the integrated system through functional and resource integration.

- This study provides a new application of ES, that is, utilising knowledge-based ES to select the most efficient MODM method for each particular decision maker (or decision group) in a particular decision problem.

- The complete method management functions of IMOGDSS guide the decision makers to use the most suitable method to solve their decision making problem, allow them to use multiple methods to resolve complex problems that could not otherwise be solved with a single MODM, and also allow them in a decision group environment to get solutions from different methods.

- This study provides a scheme to suggest and apply the best aggregation of individual solutions in MODM through integrating various GDM methods in a methodology base. This work not only deal with the different existing GDM approaches to aggregation, it also provides new aggregation methods to suit the IMOGDSS.

- This study implements a new configuration of group decision support software that provides GUI-based hierarchical procedure for solving MODM problems with intelligent guidance in a group. The procedure consists of two levels, an individual MODM level and a group aggregation level. This configuration provides a series of advantages.

10.4 Future Enhancements

The future research on IMOGDSS can be directed in many areas. Some of them are listed below:

- The knowledge-based guidance to MODM in individuals can be further expanded to deal with group aggregation process to guide GDM method selection in order to reach a compromise solution in a decision group.

- Furthermore, it would be interesting to integrate more expert systems in IMOGDSS to support the identification, analysis, formulation, evaluation and solution of MODM by a group in a user-friendly computing environment under more intelligent guidance.

- The growing popularity and availability of Internet are making it possible to create effective DSS and GDSS that are applicable to Internet (Goul 1997; Huang 1999). This study can be expanded and implementation by using the World Wide Web (WWW) infrastructure and a client/server decomposition model. Current business is undergoing a major paradigm shift that is taking it from traditional management into a world of agility. An agile corporation should be able to rapidly respond to the market changes (Huang 1999). Web-based IMOGDSS may support solutions for the decision makers who are geographically separated and operated on differing computer platforms. Therefore, one of the possible future enhancements is to implement the Web-based IMOGDSS through the WWW regardless of the geographical constraints and platforms used. Consequently, GDM is likely to become more prevalent, even if decision makers are separated by time and distance barriers. The group decision support may become an 'anywhere' and 'anytime' type of flexible organisational process. Web-Based DSS deliver decision support information or decision support models to a manager, business analyst, customer or supplier using a "thin-client" Web browser like Netscape Navigator or Internet Explorer.

- Overall, IMOGDSS will provide the means for a group of decision makers to utilise the business knowledge hidden in large databases and in group members mind in order to conduct co-operative and collaborative decision making.
- IMOGDSS may be applied in many real world decision problems and thus pure applied research can be conducted.
- For solving unconstrained MODM problems, new MODM methodologies such as the evolutionary algorithm (Binh, 1999) could be included in the MODM methodology base of IMOGDSS.

However, many challenges remain ahead, and many fruitful research can be conducted by using and enhancing IMOGDSS.

Reference

Ackermann, F. & Eden, C. 1994, 'Issues in computer and non-computer supported GDSS', *Decision Support Systems,* vol. 12, no. 4-5, pp. 381-390.

Aczel, J. S. 1983, 'Procedures for synthesizing ratio judgements', *Journal of Mathematical Psychology,* vol. 27, no. 1, pp. 93-102.

Adams, R., Bloor, I., Collier, M., Meldrum, M. & Ward, S. 1993, *Decision support systems and performance assessment in academic libraries,* Bowker Saur Ltd., London.

Adizes, I. & Turban, E. 1985, 'An Innovative approach to group decision making', *Personnel,* vol. 62, no. 4, pp. 45-49.

Agrell, P. J. 1995, 'A multicriteria framework for inventory control', *International Journal of Production Economics,* vol. 41, no. 1-3, pp. 59-70.

Ahituv, N. & Spiegler, I. 1989, 'A metalanguage design for sessions with group decision support systems', *Comp. Sci. Econ. Manage.,* vol. 2, no. 2, pp. 151-169.

Aiken, M., Hwang, C., Demagalhaes, R. & Martin, J. 1993, 'A comparison of malaysian and american groups using a group decision support system', *Journal of Information Science,* vol. 19, no. 6, pp. 489-491.

Aiken, M., Kim, D., Hwang, C. & Lu, L. C. 1995, 'A korean group decision support system', *Information Management,* vol. 28, no. 5, pp. 303-310.

Aiken, M., Martin, J., Shirani, A. & Singleton, T. 1994a, 'A group decision support system for multicultural and multilingual communication', *Decision Support Systems,* vol. 12, no. 2, pp. 93-96.

Aiken, M. W., Martin, J. S., Paolillo, J. G. P. & Shirani, A. I. 1994b, 'A group decision support system for multilingual groups', *Information and Management,* vol. 26, no. 3, pp. 155-161.

Akoka, J. 1981, 'A framework for decision support systems evaluation', *Information & Management, Amsterdam,* vol. 4, no. 3, pp. 133-142.

Alavi, M. & Napier, H. A. 1984, 'An experiment in applying the adaptive design approach to DSS development', *Information & Management, Amsterdam,* vol. 7, no. 1, pp. 21-29.

An, L. T. H., Tao, P. D. & Muu, L. D. 1996, 'Numerical solution for optimization over the efficient set by dc optimization algorithms', *Operations Research Letters,* vol. 19, no. 3, pp. 117-128.

Angehrn, A. A. & Jelassi, T. 1990, 'Supporting multicriteria decision making: New perspectives and new systems', in *ISDS Conference*, pp. 521-535.

Angehrn, A. A. & Jelassi, T. 1994, 'DSS research and precice in perspective', *Decision Support Systems,* vol. 12, no. 4,5, pp. 267-275.

Angehrn, A. A. & Luthi, H. J. 1990, 'Intelligent decision support systems: A visual interactive approach', *Interfaces,* vol. 20, no. 6, pp. 17-28.

Anonymous 1988, 'Group decision making: Approaches to problem solving', *Small Business Report,* vol. 13, no. 7, pp. 30-33.

Anonymous 1999, 'Training and development', *PEM: Plant Engineering and Maintenance,* vol. 23, no. 3, p. 32.

Anselin, L. & Arias, E. G. 1983, 'A multi-criteria framework as a decision support system for urban growth management applications: Central city redevelopment', *European Journal of Operational Research,,* vol. 13, no. 3, pp. 300-310.

Antunes, C. H., Almeida, L. A., Lopes, V. & Climaco, J. N. 1994, 'A decision support system dedicated to discrete multiple criteria problems', *Decision Support Systems,* vol. 12, no. 4-5, pp. 327-335.

Arinze, B. 1991, 'A contingency model of DSS development methodology', *Journal of MIS,* vol. 8, no. 1, pp. 149-166.

Arkad, K., Gao, X. M. & Ahlfeldt, H. 1995, 'Query-handling in mlm-based decision support systems', *Medical Informatics,* vol. 20, no. 3, pp. 229-240.

Athappilly, K. 1985, 'Successful decision making starts with DSS evaluation', *Data Management,* vol. 23, no. 2, pp. 24-28.

Avgerou, C. 1993, 'Information systems for development planning', *International Journal of Information Managemen,* vol. 13, no. 4, pp. 260-273.

Badri, M. A. 1996, 'A two-stage multiobjective scheduling model for \faculty-course-time\ assignments', *European Journal of Operational Research,* vol. 94, no. 1, pp. 16-28.

Bahl, H. C. & Hunt, R. G. 1984, 'A framework for systems analysis for decision support systems', *Information & Management, Amsterdam,* vol. 7, no. 3, p. 121.

Balthazard, P., Ferrell, W. R. & Aguilar, D. L. 1998, 'Influence allocation methods in group decision support systems', *Group Decision & Negotiation,* vol. 7, no. 4, pp. 347-362.

Barclay, S. 1987, *A user's manual to HIVIEW,* London School of Economics, London.

Barclay, S. 1988, *A user's manual to EQUITY,* London School of Economic, London.

Barkhi, R., Jacob, V. S., Pipino, L. & Pirkul, H. 1998, 'A study of the effect of communication channel and authority on group decision processes and outcomes', *Decision Support Systems,* vol. 23, no. 3, pp. 205-226.

Barlishen, K. D. & Baetz, B. W. 1995, 'Development of a decision support system for the planning of municipal solid waste composting and recycling programs', *Canadian Journal of Civil Engineering,* vol. 22, no. 3, pp. 637-645.

Barlow, J. F. 1991, 'Group decision making in computer project justification', *Journal of Systems Management,* vol. 42, no. 6, pp. 13-16,37.

Barr, S. & Sharda, R. 1997, 'Effictiveness of decision support systems - development or reliance effect', *Decision Support Systems,* vol. 21, no. 1, pp. 133-146.

Basch, R. 1999, 'High ajeevers: Valet-added searching from ask jeevers', *Database,* vol. 22, no. 3, pp. 28-34.

Basnet, C., Foulds, L. & Igbaria, M. 1996, 'Fleetmanager - a microcomputer-based decision support system for vehicle routing', *Decision Support Systems,* vol. 16, no. 3, pp. 195-207.

Belardo, S. & Harrald, J. 1992, 'A framework for the application of group decision support systems to problem of planning for catastrophic events', *IEEE Transactions on Engineering Management,* vol. 39, no. 4, pp. 400-411.

Belton, V. & Vickers, S. P. 1989, *Visual interactive sensitivity analyses for multiple criteria decision analysis: User manua,* SPV Software Products, UK.

Benayoun, R., de Montogolfier, J., Tergny , J. & Larichev, O. 1971, 'Linear programming with multiple objective functions: step method (STEM)', *Mathematical Programming,* vol. 1, no. 3, pp. 366-375.

Benbasat, I. & Konsynski, B. 1988, 'Introduction to special section on GDSS', *MIS Quarterly,* vol. 12, no. 4, pp. 588-590.

Benbasat, I. & Nault, B. R. 1990, 'An evaluation of empirical research in management support systems', *Decision Support Systems,* vol. 6, no. 3, pp. 203-226.

Bennett, J. L. 1983, *Building decision support systems,* Addison-Wesley Publishing Company, Inc, London.

Binh, T.T. 1999, 'A multiobjective evolutionary algorithm: The study cases', in *Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program,* Orlando, Florida, July 1999, pp. 127-128.

Blanning, R. W. & Reinig, B. A. 1999, 'Cross-impact analysis using group decision support systems: An application to the future of Hong Kong', *Futures,* vol. 31, no. 1, pp. 39-56.

Blind, K. & Grupp, H. 1999, 'Interdependencies between the science and technology infrastructure and innovation activities in German regions: Empirical findings and policy consequences', *Research Policy,* vol. 28, no. 5, pp. 451-468.

Bois, P. D., Brans, J. P., Cantraine, F. & Mareschal, B. 1989, 'Medics: An expert system for computer-aided diagnosis using the PROMETHEE multicriteria method', *European Journal of Operational Research,* vol. 39, no. 3, pp. 284-292.

Bonczek, R. H., Holsapple, C. W. & Whinston, A. B. 1980, 'Future directions for developing decision support systems', *Decision Sciences,* vol. 11, no. 4, pp. 616-631.

Bonczek, R. H., Holsapple, C. W. & Whinston, A. B. 1981, *Foundations of decision support systems,* Academic Press, New York.

Boonfalleur, L., Sokal, E., Nightingale, P. G., Peters, M., Otte, J. B., Ketelslegers, J. M., Dewilde, E. & Hasman, A. 1995, 'Utilization of laboratory resources - developments in knowledge-based ordering systems', *International Journal of Bio Medical Computing,* vol. 40, no. 1, pp. 17-30.

Borenstein, D. 1998, 'Idssflex - an intelligent dss for the design and evaluation of flexible manufacturing systems', *Journal of the Operational Research Society,* vol. 49, no. 7, pp. 734-744.

Bose, U., Davey, A. & Olson, D. 1997, 'Multi-attribute utility methods in group decision making-past applications and potential for inclusion in GDSS', *Omega-International Journal of Management Science,* vol. 25, no. 6, pp. 691-706.

Brans, J. P., Mareschal, B. & Vincke, P. H. 1986, 'How to select and how to rank projects: The PROMETHEE method', *European Journal of Operational Research,* vol. Vol 24,, no. 2, pp. 228-238.

Bryson, N. 1996, 'Group decision-making and the analytic hierarchy process: Exploring the consensus-relevant information content', *Computers and Operations Research,* vol. 23, no. 1, pp. 27-35.

Buchanan, J. T. 1994, 'An experimental evaluation of interactive mcdm methods and the decision making process', *Journal of the Operational Research Society,* vol. 45, no. 9, pp. 1050-1059.

Budgen, D. & Marashi, M. 1988, 'MDSE advisor: Knowledge-based techniques applied to software design assessment', *Knowledge-Based Systems,* vol. 1, no. 4, pp. 235-239.

Buede, D. M. 1992, 'Superior design features of decision analytic software', *Computers & Operations Research,* vol. 19, no. 1, pp. 43-57.

Bui, T. 1987, 'CO-OP: A decision support system for cooperative multiple criteria group decision making', *Dissertation Abstracts International Part B: Science and Engineering,* vol. 48, no. 2, p. 258.

Bui, T. X., Jelassi, T. M. & Shakun, M. F. 1990, 'Group decision and negotiation support systems (GDNSS)', *European Journal of Operational Research,* vol. 46, no. 1, pp. 141-142.

Bui, X. T. 1984, 'Building effective multiple criteria decision support systems', *Objectives, Solutions,* vol. 4, no. 1, pp. 3-16.

Bui, X. T. & Sivasankaran, T. R. 1988, 'An intelligent front end for MCDM based decision support systems', in *Proceedings of the VIIIth International Conference on Multiple Criteria Decision Methods,* Manchester, England.

Cardenosa, J., Juarez, C. & Pastor, G. 1998, 'An intelligent system for problem analysis in organizations', *Expert Systems with Applications,* vol. 15, no. 3-4, pp. 223-233.

Carmel, E., Herniter, B. & Nunamaker, J. F. 1993, 'Labor-management contract talks in an electronic meeting room: A case study', *Group Decision and Negotiation,* vol. 2, no. 1, pp. 27-60.

Chaudhry, S. S., Salchenberger, L. & Beheshtian, M. 1996, 'A small business inventory dss - design, development, and implementation issues', *Computers & Operations Research,* vol. 23, no. 1, pp. 63-72.

Chen, M. 1989, 'The integration of organization and information system modeling: A metasystem approach to the generation of group decision support systems and computer-aided software engineering', *Dissertation Abstracts International Part B: Science and Engineering,* vol. 49, no. 9, p. 208.

Chen, Z. 1998, 'Toward a better understanding of idea processors', *Information & Software Technology,* vol. 40, no. 10, pp. 541-553.

Chi, S.-C., Benjamin, C. O. & Riordan, C. A. 1996, 'Neuro-computing group decision support system for identifying facility location', in *The 1996 IEEE International*

Conference on Systems, Man and Cybernetics, IEEE, PISCATAWAY, NJ (USA), Beijing, China, pp. 2059-2064.

Chi, S.-C., Riordan, C. A. & Benjamin, C. O. 1996, 'Development and empirical study of a knowledge-based site-selection group decision support system', in *The 1996 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, PISCATAWAY, NJ (USA), Beijing, China, pp. 2053-2058.

Choi, Y. S. & Kim, S. H. 1994, 'An improved multiple criteria visual interactive method with stepwise focused contouring of efficient criterion vectors', *Computers & Operations Research*, vol. 21, no. 4, pp. 369-384.

Chu, C. H. 1989, 'Design issues for intelligent microcomputer-based decision support systems', *Journal of Information Systems Management*, vol. 6, no. 2, pp. 25-33.

Chung, I. K. & Adams, C. R. 1997, 'A study on the characteristics of group decision making behavior: Cultural difference perspective of Korea vs. U.S', *Journal of Global Information Management*, vol. 5, no. 3, pp. 18-29.

Climaco, J., Henggeler Antunes, C., Gomes Martins, A. & Traca Almeida, A. 1995, 'Multiple objective linear programming model for power generation expansion planning', *International Journal of Energy Research*, vol. 19, no. 5, pp. 419-432.

Colorni, A., Laniado, E. & Muratori, S. 1999, 'Decision support systems for environmental impact assessment of transport infrastructures', *Transportation Research, Part D: Transport & Environment*, vol. 4, no. 1, pp. 1-11.

Colson, G. & Mareschal, B. 1994, 'JUDGES: A descriptive group decision support system for the ranking of items', *Decision Support Systems*, vol. 12, no. 4,5, pp. 391-404.

Coman, A. 1996, 'Competence, power and conflict in group decision making', *Human Systems Management*, vol. 15, no. 4, pp. 245-255.

Conlon, S. P., Reithel, B. J., Aiken, M. W. & Shirani, A. I. 1994, 'A natural language processing based group decision support system', *Decision Support Systems*, vol. 12, no. 3, pp. 181-188.

Connell, N. A. D. & Powell, P. L. 1990, 'A comparison of potentialapplications of expert systems and decision support systems', *Journal of the Operational Research Society*, vol. 41, no. 5, pp. 431-439.

Cood, W. D. & Seifoed, L. M. 1978, 'Priority rankng and consensus formation', *Management Science*, vol. 24, no. 16, pp. 1721-1732.

Cook, G. J. & Swain, M. R. 1993, 'A computerized approach to decision process tracing for decision support system design', *Decision Sciences,* vol. 24, no. 5, pp. 931-952.

Cook, W. D. & Kress, M. 1985, 'Ordinal ranking with intensity of preference', *Management Science,* vol. 31, no. 1, pp. 26-32.

Corfman, K. P. & Lehmann, D. R. 1987, 'Models of cooperative group decision-making and relative influence: An experimental investigation of family purchase decisions', *Journal of Consumer Research,* vol. 14, no. 1, pp. 1-13.

Costea, I. 1993, 'IMM-DSS: intelligent multimedia decision support systems', in *The 1993 International Conference on Systems, Man and Cybernetics,* IEEE, PISCATAWAY, NJ, (USA), Le Touquet. Fr, pp. 169-172.

Couzens, A., Skitmore, M., Thorpe, T. & McAffer, R. 1996, 'Strategic information requirements in contract bidding - framework for development of a decision support system', *Civil Engineering Systems,* vol. 13, no. 2, pp. 121-139.

Crossland, M. D., Wynne, B. E. & Perkins, W. C. 1995, 'Spatial decision support systems - an overview of technology and a test of efficacy', *Decision Support Systems,* vol. 14, no. 3, pp. 219-235.

Csaki, P., Rapcsak, T., Turchanyi, P. & Vermes, M. 1995, 'R and D for group decision aid in Hungary by WINGDSS, a Microsoft Windows based group decision support system', *Decision Support Systems,* vol. 14, no. 3, pp. 205-217.

Czyzak, P. & Jaszkiewicz, A. 1998, 'Pareto simulated annealing--a metaheuristic technique for multiple-objective combinatorial optimisation', *Journal of Multi-Criteria Decision Analysis,* no. 7, pp. 34-47.

Daily, B., Whatley, A., Ash, S. R. & Steiner, R. L. 1996, 'The effects of a group decision support system on culturally diverse and culturally homogeneous group decision making', *Information and Management,* vol. 30, no. 6, pp. 281-289.

Davey, A. & Olson, D. 1998, 'Multiple criteria decision making models in group decision support', *Group Decision & Negotiation,* vol. 7, no. 1, pp. 55-75.

Davies, M. A. P. 1994, 'A multicriteria decision model application for managing group decisions', *Journal of the Operational Research Society,* vol. 45, no. 1, pp. 47-58.

Davis, J. G. & Sundaram, D. 1995, 'Petaps - a prototype decision support system for consumer product marketing and promotion', *European Journal of Operational Research,* vol. 87, no. 2, pp. 247-256.

Degoulet, P., Fieschi, M. & Chatellier, G. 1995, 'Decision support systems from the standpoint of knowledge representation', *Methods of Information in Medicine,* vol. 34, no. 1-2, pp. 202-208.

Dejong, J., Vanbuuren, J. T. & Luiten, J. P. A. 1996, 'Systematic approaches in water management - aquatic outlook and decision support systems combining monitoring, research, policy analysis and information technology', *Water Science & Technology,* vol. 34, no. 12, pp. 9-16.

Dennis, A. R. 1996, 'Information exchange and use in group decision making: You can lead a group to information, but you can't make it think', *MIS Quarterly,* vol. 20, no. 4, pp. 433-457.

Dennis, A. R., George, J. F., Jessup, L. M., Munamaker, J. F. & Vogel, D. R. 1988, 'Information technology to support group meeting', *MIS Quarterly,* vol. 12, no. 4, pp. 591-624.

Dennis, A. R., Nunamaker, J. F. & Paranka, D. 1991, 'Supporting the search for competitive advantage', *Journal of MIS,* vol. 8, no. 1, pp. 5-36.

DeSanctis, G. & Gallupe, R. B. 1985, 'Group decisiob support systems: A new frontier', *Data Base,* vol. 16, no. 4, pp. 3-10.

DeSanctis, G. & Gallupe, R. B. 1987, 'A foundation for the study of group decision support systems', *Management Science,* vol. 33, no. 5, pp. 589-609.

Despres, S. & Rosenthal, S. C. 1992, 'Designing decision support systems and expert systems with a better end-use involvement: A promising approach', *European Journal of Operational Research,* vol. 61, no. 1,2, pp. 145-153.

Dewan, R. M. & Hansen, S. C. 1994, 'Role of user capability and incentives in group and individual decision support systems: An economics perspective', *Decision Support Systems,* vol. 12, no. 1, pp. 25-40.

Dhillon, J. S., Parti, S. C. & Kothari, D. P. 1995, 'Multiobjective decision making in stochastic economic dispatch', *Electric Machines & Power Systems,* vol. 23, no. 3, pp. 289-301.

Djukanovic, M., Babic, B., Milosevic, B., Sobajic, D. J. & Pao, Y. H. 1996, 'Fuzzy linear programming based optimal fuel scheduling incorporating blending/transloading facilities', *IEEE Transactions on Power Systems,* vol. 11, no. 2, pp. 1017-1023.

Dos Santos, B. L. & Bariff, M. L. 1988, 'A study of user interface aids for model oriented decision support systems', *Management Science,* vol. 34, no. 4, pp. 15-24.

Driskell, J. E. & Salas, E. 1991, 'Group decision making under stress', *Journal of Applied Psychology,* vol. 76, no. 3, pp. 473-478.

Dunn, T. J. 1984, 'Development of decision support systems and a future outlook', *Massachusetts CPA Review,* vol. 58, no. 30-32, p. 30.

Easley, R. F. & Mackay, D. B. 1995, 'Supporting complex group decisions - a probabilistic multi-dimensional scaling approach', *Mathematical & Computer Modelling,* vol. 21, no. 12, pp. 25-33.

Eden, C. (ed.) 1989, *Strategic options development and analysis (SODA),* Chichester, Wiley.

Eden, C. & Ackermann, F. 1996, 'Horses for courses - a stakeholder approach to the evaluation of GDSS', *Group Decision and Negotiation,* vol. 5, no. 4, pp. 501-519.

Edwards, G., Kang, B. H., Preston, P. & Compton, P. 1995, 'Prudent expert systems with credentials - managing the expertise of decision support systems', *International Journal of Bio Medical Computing,* vol. 40, no. 2, pp. 125-132.

Edwards, J. S. 1992, 'Expert systems in management and administration - are they really different from decision support systems?', *European Journal of Operational Research,* vol. 61, no. 1,2, pp. 114-121.

Edwards, J. S. 2000, 'An analysis of expert systems for business decision making at different levels and in different roles', *European Journal of Information Systems,* Vol. 9, no. 1, pp. 36–42.

Edwards, W. 1977, 'How to use multivariate utility measurement for social decision making', *IEEE Trans on Systems, Man and Cybernetics,* vol. SMC-7, no. 5, pp. 326-340.

Eilon, S. 1972, 'Goals and constraints in decision making', *Operational Research Quarterly,* vol. 23, no. 1, pp. 3 - 15.

Ellis, C. A., Rein, G. L. & Jarvenpaa, S. L. 1990, 'Nick experimentation: Selected results concerning effectiveness of meeting support technology', *Journal of Management Information Systems,* vol. 6, no. 3, pp. 7-24.

Emrich, M. M. 1985, 'Artificial intelligence: Software for expert systems (part 3)', *Manufacturing Systems,* vol. 3, no. 5, pp. 38-42.

Eom, H. B. 1989, 'The current state of multiple criteria decision support systems', *Human Systems Management,* vol. 8, no. 2, pp. 113 - 119.

Eom, H. B., Lee, S. M., Snyder, C. A. & Ford, F. N. 1987, 'A multiple criteria decision support system for global financial planning', *Journal of Management Information Systems,* vol. 4, no. 3, pp. 94-113.

Eom, S. B. 1996; 'Mapping the intellectual structure of research in decision support systems through author cocitation analysis (1971-1993)', *Decision Support Systems,* vol. 16, no. 4, pp. 315-338.

Eom, S. B. 1998, 'Relationships between the decision support system subspecialties and reference disciplines: An empirical investigation', *European Journal of Operational Research,* vol. 104, no. 1, pp. 31-45.

Eom, s. B., Lee, S. M. & Kim, J. K. 1993, 'The intellectural structure of decision support systems (1971-1989)', *Decision Support System,* vol. 10, no. June, pp. 19-35.

Eom, S. B. & Min, H. 1992, 'The changing role of multiple criteria in decision support systems', *Human Systems Management,* vol. 11, no. 3, pp. 137 - 144.

Er, M. C. & Ng, A. C. 1995, 'The anonymity and proximity factors in group decision support systems', *Decision Support Systems,* vol. 14, no. 1, pp. 75-83.

Espinasse, B. & Nabitz, S. 1993, 'Distributed artificial intelligence and decision support systems: A multi-agents system for professional check-up', *Proc IEEE Int Conf Syst Man Cybern, IEEE,* vol. 1, pp. 453-458.

Everitt, B. S. 1998, *The Cambridge dictionary of statistics,* Cambridge University Press, Cambridge, England.

Expert-Choice-Inc 1992, *Expert choice: Walk-through,* Pittsburgh, USA.

Farhoomand, A. F. 1987, 'Scientific progress of management information systems', *Data Base,* vol. 18, no. Summer, pp. 48-56.

Fazlollahi, B., Parikh, M. A. & Verma, S. 1997, 'Adaptive decision support systrms', *Decision Support Systems,* vol. 20, no. 4, pp. 297-315.

Finlay, P. N. 1990, 'Decision support systems and expert systems: A comparison of their components and design methodologies', *Computers and Operations Research,* vol. 17, no. 6, pp. 535-543.

Ford, F. N. 1985, 'Decision support systems and expert systems: A comparison', *Information and Management,* vol. 8, no. 1, pp. 21-26.

Fordyce, K. J. & Sullivan, G., A. 1986, 'Decision simulation (DSIM)-- one outcome of combining expert systems and decision support systems', in *Artificial Intelligence in*

Economics and Management, ed. P. L.F., Elsevier science publishers B. V., North-Holland.

Freimer, M. & Yu, P. 1976, 'Some new results on compromise solutions for group decision problems', *Management Science,* vol. 22, no. 6, pp. 688-693.

Friend, J. & Hickling, A. 1987, *Planning under pressure: The strategic choice approach,* Oxford: Pergamon Press.

Galliers, R. D. (ed.) 1990, *Choosing appropriate information systems research approaches: A revised taxonomy,* Elsevier Science B.V. (North-Holland), Copenhagen, Denmark.

Gallupe, R. B., DeSanctis, G. & Dickson, G. W. 1988, 'Computer-based support for group problem finding: An experimental investigation', *MIS Quarterly,* vol. 12, no. 2, pp. 277-296.

Gallupe, R. B. & McKeen, J. D. 1990, 'Enhancing computer-mediated communication: An experimental investigation into the use of a group decision support system for face-to-face versus remote meetings', *Information and Management,* vol. 18, no. 1, pp. 1-13.

Gan, R. C. & Yang, D. 1994, 'Case-based decision support system with artificial neural network', *Computers & Industrial Engineering,* vol. 27, no. 1-4, pp. 437-440.

Garsombke, H. P. & Parker, L. M. 1987, 'Decision support systems and expert systems: Auditing in the information age', *Journal of Accounting and EDP,* vol. 2, no. 4, pp. 20-25.

Gavish, B., Gerdes, J. & Sridhar, S. 1995, 'Cm(3) - a distributed group decision support system', *IIE Transactions,* vol. 27, no. 6, pp. 722-733.

Gavish, B. & Gerdes, J. H. 1998, 'Anonymous mechanisms in group decision support systems communication', *Decision Support Systems,* vol. 23, no. 4, pp. 297-328.

Gen, M., Ida, K., Tsujimura, Y. & Kim, C. E. 1993, 'Large-scale 0-1 fuzzy goal programming and its application to reliability optimization problem', *Computers & Industrial Engineering,* vol. 24, no. 4, pp. 539-549.

Goicoechea, A. 1989, 'On the use of multiple attributes and imperfect information with ARIADNE to rank alternatives', *Applied Mathematics &Computation Journal,* vol. 63, no. 1, pp. 67-80.

Golden, B., Wasil, E. & Harker, P. 1990, *The analytic hierarchy process: Applications and studies,* Springer Verlag,, New York.

Gottinger, H. W. & Weimann, H. P. 1995, 'Intelligent inference systems based on influence diagrams', *Decision Support Systems,* vol. 15, no. 1, pp. 27-43.

Goul, M., Philippakis, A., Kiang, M., Fernandes, D. & Otondo, R. 1997, 'Requirements for the design of a protocol suite to automate DSS development on the World Wide Web: A client/server approach', *Decision Support Systems,* vol. 19, no. 3, pp. 151-170.

Gray, P. 1987, 'Group decision support systems', *Decision Support Systems,* vol. 3, no. 3, pp. 233-242.

Gray, p. & Mandviwalla, m. 1999, 'New direction for GDSS', *Group Decision and Negotiation,* vol. 8, no. 1, pp. 77-83.

Gray, P. & Olfman, L. 1989, 'The user interface in group decision support systems', *Decision Support Systems,* vol. 5, no. 2, pp. 119-137.

Grohowski, R. B., McGoff, C., Vogel, D. R., Martz, W. B. & Nunamaker, J. F. 1990, 'Implementation of electronic meeting systems at IBM', *MIS Quarterly,* vol. 14, no. 4, pp. 369-383.

Guba, E. G. 1990, *The alternative paradigm dialog,* Newbury Park,, CA: Sage.

Hai, Z. G. 1998, 'Conflict group decision training: model and system', *Knowledge Based Systems,* vol. 11, no. 3-4, pp. 191-196.

Hammond, D. H., Clark, T. D., Hartman, S. J. & Crow, S. M. 1995, 'One more time - does a computer-based decision support system improve managerial decisions', *International Journal of Computer Applications in Technology,* vol. 8, no. 5-6, pp. 290-300.

Hartman, S. J. & Nelson, B. H. 1996, 'Group decision making in the negative domain', *Group and Organization Management,* vol. 21, no. 2, pp. 146-162.

Heaney, J. P., Wright, L. & Sample, D. 1999, 'Research needs in urban wet weather flows', *Water Environment Research,* vol. 71, no. 2, pp. 241-250.

Hee, K. M. & Lenstra, J. K. 1994, 'A comparative study in DSS development', *European Journal of Operational Research,* vol. 79, no. 1, pp. 153-157.

Henderson, J. C. 1987, 'Finding synergy between decision support systems and expert systems research', *Decision Sciences,* vol. 18, no. 3, pp. 333-349.

Herriot, P., Chalmers, C. & Wingrove, J. 1985, 'Group decision making in an assessment centre', *Journal of Occupational Psychology,* vol. 58, no. 4, pp. 309-312.

Higby, M. A. & Farah, B. N. 1991, 'The status of marketing information systems, decision support systems and expert systems in the marketing function of U.S. firms', *Information and Management,* vol. 20, no. 1, pp. 29-35.

Hiltz, S. R., Turoff, M. & Johnson, K. 1989, 'Experiments in group decision making, 3: Disinhibition, deindividuation, and group process in pen name and real name computer conferences', *Decision Support Systems,* vol. 5, no. 2, pp. 217-232.

Hipel, K. 1992, 'Multiple objective decision making in water resources', *Water Resources Bulletin,* vol. 28, no. 1, pp. 3-12.

Hipel, K. W., Kilgour, D. M., Fang, L. & Peng, X. 1998, 'Conflict management for the service industries', in *IEEE International Conference on Systems, Man and Cybernetics,* pp. 4769-4775.

Ho, J. L. Y. 1999, 'Technology and group decision process in going-concern judgements', *Group Decision & Negotiation,* vol. 8, no. 1, pp. 33-49.

Hochman, Z., Mears, P. T., Farquharson, R. J., Hindmarsh, J. H. & Pearson, C. J. 1995, 'A multiple-domain knowledge based system - coordination and integration of agronomic and economic knowledge bases with databases and bio-economic models', *Agricultural Systems,* vol. 48, no. 2, pp. 141-161.

Hochman, Z., Pearson, C. J. & Litchfield, R. W. 1994, 'Users attitudes and roles in the development and evaluation of knowledge based decision support systems for agricultural advisers', *Agricultural Systems,* vol. 44, no. 2, pp. 217-235.

Holsapple, C. W., Johnson, L. E., Manakyan, H. & Tanner, J. 1995, 'An empirical assessment and categorization of journals relevant to dss research', *Decision Support Systems,* vol. 14, no. 4, pp. 359-367.

Holsapple, C. W. & Whinston, A. B. 1987, 'Decision support systems: Theory and applications', *Springer Verlag NATO ASI Series, Computer and System Science,* vol. 31, no. (procedings of the NATO), pp. 34-42.

Horst, R. & Thoai, N. V. 1997, 'Utility function programs and optimization over the efficient set in multiple-objective decision making', *Journal of Optimization Theory & Applications,* vol. 92, no. 3, pp. 605-631.

Huang, C. 1999, 'An agile approach to logical network analysis in decision support systems', *Decision Support Systems,* vol. 25, no. 3, pp. 53-70.

Huber, G. P. 1984, 'Issues in the design of group decision support systems', *MIS Quarterly,* vol. 8, no. 3, pp. 195-204.

Hudson, D. L., Cohen, M. E. & Deedwania, P. C. 1994, 'Information integration in a decision support system', *Journal of the American Medical Informatics Association,* no. Suppl S, p. 1051.

Hughes, A. & Grawoig, D. E. 1971, *Statistics: a foundation for analysis,* Addison-Wesley Pub. Co.

Hwang, C. & Lin, M. 1987, *Group decision making under multiple criteria,* Springer Verlag.

Hwang, C. L., Lai, Y. J. & Liu, T. Y. 1993, 'A new approach for multiple objective decision making', *Computers & Operations Research,* vol. 20, no. 8, pp. 889-899.

Hwang, C. L. & Masud, A. S. M. 1979, *Multiple objective decision making - methods and applications: A state of the art survey,* Springer, New York.

Hwang, C. L. & Yoon, K. 1981, *Multiple attribute decision making - a state of the art survey,* Springer, Berlin.

Hwang, C.-L., Lai, Y.-J. & Ko, M.-D. 1993, 'ISGP-II for multiobjective optimization with imprecise objective coefficien', *Computers & Operations Research,* vol. 20, no. 5, pp. 503-514.

Hwang, M. 1998, 'Did task type matter in the use of decision room GSS - a critical review and a meta-analysis', *Omega-International Journal of Management Science,* vol. 26, no. 1, pp. 1-15.

Igbaria, M. & Guimaraes, T. 1994, 'Empirically testing the outcomes of user involvement in dss development', *Omega International Journal of Management Science,* vol. 22, no. 2, pp. 157-172.

Ignizio, J. P. 1976, *Goal programming and extensions,* Massachusetts.

Ignizio, J. P. 1981, 'The determination of a subset of efficient solutions via goal programming', *Computer and Operations Research,* vol. 8, no. 1, pp. 9-16.

Iselin, E. R. 1991, 'Individual versus group decision-making performance: A further investigation of two theories in a bankruptcy prediction task', *Journal of Business Finance and Accounting,* vol. 18, no. 2, pp. 191-208.

Islei, G., Lockett, G. & Naude, P. 1999, 'Judgemental modelling as an aid to scenario planning and analysis', *Omega International Journal of Management Science,* vol. 27, no. 1, pp. 61-73.

Iz, P. & Krajewski, L. 1992, 'Comparative evaluation of three interactive multiobective programming techniques as group decision support tools', *INFOR,* vol. 30, no. 4, pp. 349-363.

Iz, P. H. 1990, 'A group decision aid based on the tchebycheff procedure', in *The 22nd National Meeting of Decision Science Institute*, Atlanta, pp. 245-253.

Iz, P. H. 1991, 'Group decision support and multiple criteria optimization', in *Proceedings of the Hawaii International Conference on System Sciences*, Hawaii, pp. 678-686.

Iz, P. H. 1992a, 'An experimental assessment of preference aggregation in a group decision system based on multiple criteria optimization', in *Proceedings of the Hawaii International Conference on System Sciences,* Hawaii, pp. 185-189.

Iz, P. H. 1992b, 'Two multiple criteria group decision support systems based on mathematical programming and ranking methods', *European Journal of Operational Research,* vol. 61, pp. 245-253.

Iz, P. H. & Gardiner, L. R. 1993, 'Analysis of multiple criteria decision support systems for cooperative groups', *Group Decision and Negotiation,* vol. 2, no. 1, pp. 61-79.

Iz, P. H. & Jelassi, M. T. 1990, 'An interactive group decision aid for multiobjective problems: An empirical assessment', *OMEGA International Journal of Management Science,* vol. 18, no. 6, pp. 295-604.

Iz, P. H. & Korhonen, P. 1998, 'Special issue on group decision and negotiation based on multiple criteria - introduction', *Group Decision & Negotiation,* vol. 7, no. 1, pp. 1-2.

Jacob, V. S. & Pirkul, H. 1992, 'A framework for supporting distributed group decision-making', *Decision Support Systems,* vol. 8, no. 1, pp. 17-28.

Jacucci, G., Foy, M. & Uhrik, C. 1996a, 'Developing transportable agricultural decision support systems .1. a conceptual framework', *Computers & Electronics in Agriculture,* vol. 14, no. 4, pp. 291-300.

Jacucci, G., Foy, M. & Uhrik, C. 1996b, 'Developing transportable agricultural decision support systems .2. an example', *Computers & Electronics in Agriculture,* vol. 14, no. 4, pp. 301-315.

Jang, S. Y., Park, J. & Park, N. 1996, 'An integrated decision support system for fms production planning and scheduling problems', *International Journal of Advanced Manufacturing Technology,* vol. 11, no. 2, pp. 101-110.

Jarke, M., Jelassi, M. T. & Shakun, M. F. 1987, 'Mediator: Towards a negotiation support system', *EJOR,* vol. 31, no. 2, pp. 314-316.

Jarvenpaa, S. L., Rao, V. S. & Huber, G. P. 1988, 'Computer support for meeting of groups working on unstructured problems: A field experiment', *MIS Quarterly,* vol. 12, no. 4, pp. 645-666.

Jelassi, M. T. & Beauclair, R. A. 1987, 'An integrated framework for group decision support design', *Information & Management,* vol. 13, no. 3, pp. 143-153.

Jelassi, M. T., Jarke, M. & Stohr, E. A. 1985, 'Designing a generalized multiple criteria decision support system', *Journal of Management Information Systems,* vol. 1, no. 4, pp. 24-43.

Jessup, L. M. & Tansik, D. A. 1991, 'Decision making in an automated environment: The effects of anonymity and proximity with a group decision support system', *Decision Sciences,* vol. 22, no. 2, pp. 266-279.

Jiang, J. J., Muhanna, W. A. & Pick, R. A. 1996, 'The impact of model performance history information on users confidence in decision models - an experimental examination', *Computers in Human Behavior,* vol. 12, no. 2, pp. 193-207.

Johnson, V. 1991, 'Group decision making', *Successful Meetings,* vol. 40, no. 7, pp. 76-77.

Kahn, C. E. 1993, 'Graphical knowledge presentation in a mumps-based decision-support system', *Computer Methods & Programs in Biomedicine,* vol. 40, no. 3, pp. 159-166.

Kananen, I., Korhonen, P., Wallenius, J. & Wallenius, H. 1990, 'Multiple objective analysis of input-output models for emergency management', *Operations Research,* vol. 38, no. 2, pp. 193-201.

Karacapilidis, N. I. & Pappis, C. P. 1996, 'Production planning and control in textile industry - a case study', *Computers in Industry,* vol. 30, no. 2, pp. 127-144.

Karacapilidis, N. I. & Pappis, C. P. 1997, 'A framework for group decision support systems: Combining AI tools and OR techniques', *European Journal of Operational Research,* vol. 103, no. 2, pp. 373-388.

Karan, V., Kerr, D. S., Murthy, U. S. & Vinze, A. S. 1996, 'Information technology support for collaborative decision making in auditing: An experimental investigation', *Decision Support Systems,* vol. 16, no. 3, pp. 181-194.

Kasanen, E., Ostermark, R. & Zeleny, M. 1991, 'Gestalt system of holistic graphics: New management support view of MCDM', *Computers & Operations Research,* vol. 18, no. 2, pp. 233-239.

Kasper, G. M. 1996, 'A theory of decision support system design for user calibration', *Information Systems Research,* vol. 7, no. 2, pp. 215-232.

Kawamura, H. 1993, 'Fuzzy network for decision support systems', *Fuzzy Sets & Systems,* vol. 58, no. 1, pp. 59-72.

Keen, P. G. W. 1976, 'Interactive computer systems for managers: A modest proposal', *Sloan Management Review,* vol. 18, no. 1, pp. 1-17.

Keen, P. G. W. 1980, 'Adaptive design for decision support systems', *Data Base,* vol. 12, no. 1, pp. 15-25.

Keen, P. G. W. 1987, 'Decision support system: The next decade', *Decision Support System,* vol. 3, no. 3, pp. 253-265.

keeney, R. L. 1988, 'Value-driven expert systems for decision support', *Decision Support Systems,* vol. 4, no. 4, pp. 405-412.

Keeney, R. L. & Raiffa, H. 1976, *Decisions with multiple objectives: Preferences and value tradeoff,* Wiley, New York.

Kerr, N. L. 1992, 'Group decision making at a multialternative task: Extremity, interfaction distance, pluralities, and issue importance', *Organizational Behavior and Human Decision Processes,* vol. 52, no. 1, pp. 64-95.

Kersten, G. E. 1985, 'NEGO - group decision support system', *Information and Management,* vol. 8, no. 5, pp. 237-246.

Khoong, C. M. 1995, 'Decision support systems - an extended research agenda', *Omega International Journal of Management Science,* vol. 23, no. 2, pp. 221-229.

Kiesler, S. & Sproull, L. 1992, 'Group decision making and communication technology', *Organizational Behavior and Human Decision Processes,* vol. 52, no. 1, pp. 96-123.

Kim, S. H., Ahn, B. S. & Choi, S. H. 1997, 'An efficient force planning system using multi-objective linear goal programming', *Computers & Operations Research,* vol. 24, no. 6, pp. 569-580.

Kim, S. H. & Han, C. H. 1999, 'An interactive procedure for multi-attribute group decision making with incomplete information', *Computers and Operations Research,* vol. 26, no. 8, pp. 755-772.

Kivijarvi, H., Korhonen, P. & Wallenius, J. 1986, 'Operations research and Its practice in Finland', *Interfaces,* vol. 16, no. 4, pp. 53-59.

Klein, H. K. & Hirschheim, R. A. 1985, 'Fundamental issues of decision support systems: A consequentialist perspective', *Decision Support Systems, Amsterdam,* vol. 1, no. 1, pp. 5-24.

Klein, M. R. & Methlie, L. B. 1995, *Knowledge-based decision support systems with applications in busines,* John Wiley & Sons.

Klimberg, R. 1992, 'GRADS: A new graphical display system for visualizing multiple criteria solutions', *Computers & Operations Research,* vol. 19, no. 7, pp. 707-711.

Ko, C. E. & Lin, T. W. 1988, 'Multiple criteria decision making and expert systems', in *Applied Expert Systems*, eds. E. Turban & P. R. Watkins, Elsevier, New York, pp. 33-45.

Kobbacy, K. A. H., Proudlove, N. C. & Harper, M. A. 1995, 'Towards an intelligent maintenance optimization system', *Journal of the Operational Research Society,* vol. 46, no. 7, pp. 831-853.

Kok, M. & Lootsma, F. A. 1985, 'Pairwise comparison methods in multiple objective programming with applications in a long-term energy planning model,', *European Journal of Operational research,* vol. 22, no. 1, pp. 44-55.

Korhonen, A. 1987, 'A dynamic bank portfolio planning model with multiple scenarios, multiple goals and changing piorities', *European Journal of Operational Research,* vol. 30, no. 1, pp. 13-23.

Korhonen, P. 1988, 'A visual reference direction approach to solving discrete multiple criteria problems', *European Journal of Operational Research,* vol. 34, no. 2, pp. 152-159.

Korhonen, P. & Laakso, J. 1986a, 'Solving generalized goal programming problems using a visual interactive approach', *European Journal of Operational Research,* vol. 26, no. 3, pp. 355-363.

Korhonen, P., Lewandowki, A. & Wallenius, J. (eds.) 1991, *Multiple criteria decision support,* Springer-Verlag, Berlin.

Korhonen, P., Moskowitz, H. & Wallenius, J. 1986, 'A progressive algorithm for modeling and solving multiple-criteria decision problems', *Operations Research,* vol. 34, no. 5, pp. 726-731.

Korhonen, P., Moskowitz, H. & Wallenius, J. 1992, 'Multiple criteria decision support - a review,', *European Journal of Operational Research,* vol. 63, no. 3, pp. 361 - 375.

Korhonen, P. & Soismaa, M. 1988, 'A multiple criteria model for pricing alcoholic beverages', *European Journal of Operational Research,* vol. 37, no. 2, pp. 165-175.

Korhonen, P. & Wallenius, J. 1986, 'Some theory and an approach to solving sequential multiple-criteria decision problems', *Journal of the Operational Research Society,* vol. 37, no. 5, pp. 501-508.

Korhonen, P. & Wallenius, J. 1990a, 'A multiple objective linear programming decision support system', *Decision Support Systems,* vol. 6, no. 3, pp. 243-251.

Korhonen, P. & Wallenius, J. 1990b, 'Using qualitative data in multiple objective linear programming', *European Journal of Operational Research,* vol. 48, no. 1, pp. 81-87.

Korhonen, P. J. 1986, 'A hierarchical interactive method for ranking alternatives with multiple qualitative criteria', *European Journal of Operational Research,* vol. 24, no. 2, pp. 265-276.

Korhonen, P. J. & Laakso, J. 1986b, 'A visual interactive method for solving the multiple criteria problem', *European Journal of Operational Research,* vol. 24, no. 2, pp. 277-287.

Korpela, J. & Tuominen, M. 1996, 'Inventory forecasting with a multiple criteria decision tool', *International Journal of Production Economics,* vol. 45, no. 1-3, pp. 159-168.

Krusinska, E., Babic, A., Mathiesen, U., Chowdhury, S., Wigertz, O., Bodemar, G. & Franzen, L. 1993, 'A statistically rule-based decision support system for the management of patients with suspected liver disease', *Medical Informatics,* vol. 18, no. 2, pp. 113-130.

Kull, D. 1982, 'Group decisions: Can computers help', *Comp. Decisions,* vol. 14, no. 5, pp. 70-84, 160.

Kuncheva, L. 1993, 'An aggregation of pro and con evidence for medical decision support system', *Computers in Biology & Medicine,* vol. 23, no. 6, pp. 417-424.

Lagreze, E. J. & Shakun, M. F. 1984, ' Decision support systems for semi-structured buying decisions', *European Journal of Operational Research,* vol. 16, no. 1, pp. 48-58.

Lai, Y. J. 1995, 'Imost - interactive multiple objective system technique', *Journal of the Operational Research Society,* vol. 46, no. 8, pp. 958-976.

Lai, Y. J. & Chang, S. J. 1994, 'A fuzzy approach for multiresponse optimization - an off-line quality engineering problem', *Fuzzy Sets & Systems,* vol. 63, no. 2, pp. 117-129.

Lai, Y. J., Liu, T. Y. & Hwang, C. L. 1994a, 'Topsis for MODM', *European Journal of Operational Research,* vol. 76, no. 3, pp. 486-500.

Larichev, O. I., Olson, D. L., Moshkovich, H. M. & Mechitov, A. J. 1995, 'Numerical vs cardinal measurements in multiattribute decision making - how exact is enough', *Organizational Behavior & Human Decision Processes,* vol. 64, no. 1, pp. 9-21.

Lawrence, P., Stone, J., Heilman, P. & Lane, L. 1997, 'Using measured data and expert opinion in a multiple objective decision support system for semiarid rangelands', *Transactions of the ASAE,* vol. 40, no. 6, pp. 1589-1597.

Lei, Y. & Youmin, X. 1996, 'A view of group decision making process and bivoting approach', *Computers and Industrial Engineering,* vol. 31, no. 3,4, pp. 945-948.

Leighton, R. T. 1981, 'Decision support systems', *Journal of Systems Management,,* vol. 32, no. 2, pp. 4-11.

Levine, P., Pomerol, M.-J. & Saneh, R. 1990, 'Rules integrate data in a multicriteria decision support system', *IEEE Transaction on Man, Machine and Systems,* vol. 20, no. 3, pp. 678-686.

Lewandowski, A. 1989, 'SCDAS - decision support system for group decision making: Decision theoretic framework', *Decision Support Systems,* vol. 5, no. 4, pp. 403-423.

Lewandowski, A. & Stanchev, I. (eds.) 1989, *Methodology and software for interactive decision suppor,* Springer-Verlag, Berlin.

Lewandowski, A. & Wierzbicki, A. (eds.) 1989, *Aspiration based decision support systems,* Springer-Verlag, Berlin.

Lewis, L. F. & Keleman, K. S. 1988, 'Issues in group decision support system (GDSS) design', *Journal of Information Science Principles and Practice,* vol. 14, no. 6, pp. 347-354.

Li, H. X. 1991, *The methods and applications of fuzzy mathematics,* Tianjin Science and Technology Publisher, Tianjin,China.

Liberatore, M. J. & Stylianou, A. C. 1993, 'The development managers advisory system - a knowledge-based dss tool for project assessment', *Decision Sciences,* vol. 24, no. 5, pp. 953-976.

Liberty, J. 1998, *Beginning object-oriented analysis and design with C++,* Wrox Press Ltd, Olton.

Liebowitz, J. 1988, 'Beyond decision support systems: The role of operations research in expert systems', *Computers and Industrial Engineering,* vol. 14, no. 4, pp. 415-418.

Lihovd, E., Johannssen, T. I., Steinebach, C. & Rasmussen, M. 1998, 'Intelligent diagnosis and maintenance management', *Journal of Intelligent Manufacturing*, vol. 9, no. 6, pp. 523-537.

Lim, L. H., Raman, K. S. & Wei, K. K. 1994, 'Interacting effects of gdss and leadership', *Decision Support Systems*, vol. 12, no. 3, pp. 199-211.

Limayem, M. & Chelbi, A. 1997, 'Improving multicriteria group decision making with automated decision guidance', in *The 1997 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, PISCATAWAY,NJ,(USA), Orlando,FL,USA, pp. 1890-1895.

Liou, Y. I. & Nunamaker, J. F., Jr. 1993, 'An investigation into knowledge acquisition using a group decision support system', *Information and Management*, vol. 24, no. 3, pp. 121-132.

Lobach, D. F. & Hammond, W. E. 1997, 'Computerized decision support based on a clinical practice guideline improves compliance with care standards', *American Journal of Medicine*, vol. 102, no. 1, pp. 89-98.

Lu, J., Quaddus, M. A. & Williams, R. 1999, 'A framework and prototype for intelligent multiple objectives decision support system', in *1999 Asia Pacific Decision Sciences Institute Conference*, Shanghai, China.

Maclean, D. A. 1996, 'The role of a stand dynamics model in the spruce budworm decision support system', *Canadian Journal of Forest Research Journal Canadien de la Recherche Forestiere*, vol. 26, no. 10, pp. 1731-1741.

Madu, C. 1994, 'A quality confidence procedure for GDSS application in multicriteria decision making', *IIE Transactions*, vol. 26, no. 3, pp. 31-39.

Mandviwalla, M. & Gray, P. 1998, 'Is IS research on GSS relevant?', *Information Resources Management Journal*, vol. 11, no. 1, pp. 7-15.

Marchewka, J. T. & Mark, K. 1995, 'Portfolio theory approach for selecting and managing IT projects', *Information Resources Management Journal*, vol. 8, no. 4, pp. 5-14.

Massam, B. H. 1988, 'Multi-criteria decision making (MCDM) techniques in planning', *Progress in Planning*, vol. 30, pp. 1 - 84.

Masud, A. S. M. & Hwang, C. L. 1981, 'Interactive sequential goal programming', *Journal of the Operational Research Society*, vol. 32, no. 5, pp. 391-400.

McCartt, A. T. & Rohrbaugh, J. 1989, 'Evaluating group decision support system effectiveness: A performance study of decision conferencing', *Decision Support Systems,* vol. 5, no. 2, pp. 243-253.

McIvor, R. T., Mulvenna, M. D. & Humphreys, P. K. 1997, 'A hybrid knowledge-based system for strategic purchasing', *Expert Systems with Applications,* vol. 12, no. 4, pp. 497-512.

Mentzas, G. N. 1996, 'Team coordination in decision support projects', *European Journal of Operational Research,* vol. 89, no. 1, pp. 70-85.

Milton-Smith, J., Schmidenberg, O. & Klass, D. 1999, 'Strategic planning, strategic goals and group decision support systems: A business school case study', *International Journal of Technology Management,* vol. 17, no. 1-2, pp. 173-194.

Minch, R. P. & Sanders, G. L. 1986, 'Computerized information systems supporting multicriteria decision making', *Decision Sciences,* vol. 17, no. 3, pp. 395 - 413.

Mockler, R. J. & Dologite, D. G. 1991, 'Using computer software to improve group decision-making', *Long Range Planning,* vol. 24, no. 4, pp. 44-57.

Monplaisir, L. F., Riordan, C. & Benjamin, C. O. 1999, 'Comparison of intelligent CSCW architectures for the evaluation of agile manufacturing systems designs', *International Journal of Human Factors in Manufacturing,* vol. 9, no. 2, pp. 137-150.

Montgomery, S. L. 1998, *Building object-oriented software,* McGraw-Hill, Inc., New York.

Moreno, J. A. 1994, 'On the implementation of preference relational systems in a computer-aided decision support system', *Cybernetics & Systems,* vol. 25, no. 1, pp. 17-37.

Morgan, R. F. 1996, 'An intelligent decision support system for a health authority - solving information overload', *Journal of the Operational Research Society,* vol. 47, no. 4, pp. 570-582.

Muhanna, W. A. 1993, 'Object-oriented framework for model management and DSS development', *Decision Support System,* vol. 9, no. 2, pp. 217-229.

Mukhopadhyay, S. K. 1995, 'Optimal scheduling of just-in-time purchase deliveries', *International Journal of Operations & Production Management,* vol. 15, no. 9, pp. 59-63.

Murrell, S. & Plant, R. T. 1997, 'A survey of tools for the validation and verification of knowledge-based systems: 1985-1995', *Decision Support Systems,* vol. 21, no. 4, pp. 307-323.

Nam, K., Rajagopalan, S., Rao, H. R. & Chaudhury, A. 1996, 'A two-level investigation of information systems outsourcing', *Communications of the ACM,* vol. 39, no. 7, pp. 36-44.

Nazareth, D. L. 1993, 'Integrating MCDM and DSS: Barriers and counter strategies', *Infor,* vol. 31, no. 1, pp. 1-15.

Niederman, F., Beise, C. M. & Beranek, P. M. 1996, 'Issues and concerns about computer-supported meeting: the facilitator's perspective', *MIS Quarterly,* vol. 20, no. 1, pp. 1-22.

Nogami, J., Nakasuka, S. & Tanabe, T. 1996, 'Real-time decision support for air traffic management, utilizing machine learning', *Control Engineering Practice,* vol. 4, no. 8, pp. 1129-1141.

Noori, H. 1995, 'The design of an integrated group decision support system for technology assessment', *R and D Management,* vol. 25, no. 3, pp. 309-322.

Nunamaker, J. F., Chen, M. & Purdin, T. D. M. 1990-1991, 'Systems development in information systems research', *Journal of Management Information Systems,* vol. 7, no. 3, pp. 89-106.

Ogryczak, W., Studzinski, K. & Zorychta, K. 1989, 'A solver for the multi-objective transshipment problem with facility location', *European Journal of Operational Research.,* vol. 43, no. 1, pp. 53-64.

Ong, K. L. & Lee, R. M. 1996, 'A decision support system for bureaucratic policy administration - an abductive logic programming approach', *Decision Support Systems,* vol. 16, no. 1, pp. 21-38.

Owens, H. D. & Philippakis, A. S. 1995, 'Inductive consistency in knowledge-based decision support systems', *Decision Support Systems,* vol. 13, no. 2, pp. 167-181.

Paese, P. W., Bieser, M. & Tubbs, M. E. 1993, 'Framing effects and choice shifts in group decision making', *Organizational Behavior and Human Decision Processes,* vol. 56, no. 1, pp. 149-165.

Paige, G. B., Stone, J. J., Lane, L. J. & Hakonson, T. E. 1996a, 'Calibration and testing of simulation models for evaluation of trench cap designs', *Journal of Environmental Quality,* vol. 25, no. 1, pp. 136-144.

Paige, G. B., Stone, J. J., Lane, L. J., Yakowitz, D. S. & Hakonson, T. E. 1996b, 'Evaluation of a prototype decision support system for selecting trench cap designs', *Journal of Environmental Quality,* vol. 25, no. 1, pp. 127-135.

Pakath, R. & Rao, H. R. 1991, 'Module base design for a portfolio of institutional decision support systems', *Information and Management,* vol. 20, no. 4, pp. 265-278.

Palmer, J. D., Fields, N. & Brouse, P. L. 1994, 'Multigroup decision-support systems in CSCW', *Computer,* vol. 27, no. 5, pp. 67-72.

Patel, S. A. & Kamrani, A. K. 1996, 'Intelligent decision support system for diagnosis and maintenance of automated systems', *Computers and Industrial Engineering,* vol. 30, no. 2, pp. 297-319.

Pau, L. F. 1986, 'Artificial intelligence in economics and management: Why?', in *Artificial Intelligence in Economics and Management*, ed. L. F. Pau, Elsevier science publishers B. V., North-Holland.

Pearson, J. M. & Shim, J. P. 1995, 'An empirical investigation into dss structures and environments', *Decision Support Systems,* vol. 13, no. 2, pp. 141-158.

Pendergast, M. & Hayne, S. 1999, 'Groupware and social networks: will life ever be the same again?', *Information & Software Technology,* vol. 41, no. 6, pp. 311-318.

Pereira, J. M. C. & Duckstein, L. 1993, 'A multiple criteria decision-making approach to gis-based land suitability evaluation', *International Journal of Geographical Information Systems,* vol. 7, no. 5, pp. 407-424.

Petry, F. 1990, 'Who is afraid of choices? A proposal for multi-criteria analysis as a tool for decision Making support in development planning', *Journal of International Developmen,* vol. 2, no. 2, pp. 209-231.

Phillips, L. D. (ed.) 1988, *People-centred group decision support,* Chichester, Ellis Horwood,.

Phillips, L. D. & Phillips, M. C. 1993, 'Facilitated work groups: Theory and practice', *Journal of the Operational Research Society,* vol. 44, no. 6, pp. 533-549.

Pinson, S. & Moraitis, p. 1996, 'An intelligent distributed system for strategic decision making', *Group Decision and Negotiation,* vol. 6, no. 1, pp. 77-108.

Pitty, D. L. & Reeves, P. I. 1995, 'Developing decision support systems - a change in emphasis', *Computer Methods & Programs in Biomedicine,* vol. 48, no. 1-2, pp. 35-38.

Poh, K. 1998, 'Knowledge-based guidance system for multi-attribute decision making', *Artificial Intelligence in Engineering,* vol. 12, no. 3, pp. 315-326.

Poh, K. L. & Quaddus, M. A. 1990, 'A hybrid approach to multiobjective linear optimization', *Journal of Operational Research Society,* vol. 41, no. 11, pp. 1037-1048.

Poh, K. L., Quaddus, M. A. & Chin, K. L. 1995, 'MOLP-PC: An interactive decision support environment for multiple objective linear optimization', In *OR Applications in Singapore*, ed. M. C. T. L. Goh, Operational Research Society of Singapore, , pp. 25-39.

Pomerol, J.-C. 1993, 'From aggregating by rules to the integration of expert systems in multicriteria DSSs', in *The IEEE International Conference on Systems, Man and Cyberneties*, IEEE.

Powell, P. L. & Johnson, J. E. V. 1995, 'Gender and dss design - the research implications', *Decision Support Systems,* vol. 14, no. 1, pp. 27-58.

Power, J. M. & Saarenmaa, H. 1995, 'Object-oriented modeling and GIS intergration in a decision support system for the management of eastern hemlock looper in new foundland', *Computers & Electronics in Agriculture,* vol. 12, no. 1, pp. 1-18.

Prasad, S. Y., Karwan, M. H. & Zionts, S. 1997, 'Use of convex cones in interactive multiple objective decision making', *Management Science,* vol. 43, no. 5, pp. 723-734.

Priem, R. L., Harrison, D. A. & Muir, N. K. 1995, 'Structured conflict and consensus outcomes in group decision making', *Journal of Management,* vol. 21, no. 4, pp. 691-710.

Quaddus, M. & Klass, D. 1998, 'ALLOCATE: An Interactive system for multi-criteria resource allocation', in *Optimization Techniques and Application,* ed. L. C. K L Teo, P F Siew, Y H Leung, L S Jennings and V Rehbock eds, Perth, Australia,, pp. 471-478.

Quaddus, M. A. 1993, 'Group decision and negotiation support in multiple criteria decision making: An interactive approach', in *Proceedings of the 26th Hawaii International Conference on System Sciences,* Hawaii, pp. 247-254.

Quaddus, M. A. 1996, 'Multiple criteria decision support for the strategic analysis of information system projects', *International Journal of Managemen,* vol. 13, no. 4, December 1996, pp. 103-111.

Quaddus, M. A. 1997, 'Computer aided learning of multiple criteria decision making: An integrated system and its effectiveness', *Computers and Education,* vol. 28, no. 2, pp. 103-111.

Quaddus, M. A., Atkinson, D. J. & Levy, M. 1992, 'An application of decision conferencing to strategic planning for a voluntary organization', *Decision Analysis,* vol. 22, no. 6, pp. 61-71.

Quaddus, M. A. & Holzman, A. G. 1986, 'IMOLP: An Interactive method for multiple objective linear programs', *IEEE Transactions on Systems, Man, and Cybernetics,* vol. SMC-16, no. 3, pp. 462 - 468.

Quaddus, M.A. & Siddique, M.A.B. (1996), 'Multiobjective decision support in developing and planning', In Ghosh, R.N., Melotte, Y.M. and Siddique, M.A.B. (eds.), *Economic Development and Change*, New Age International Publishers, New Delhi/London, pp. 68-80.

Radermacher, F. J. 1994, 'Decision support systems - scope and potential', *Decision Support Systems,* vol. 12, no. 4-5, pp. 257-265.

Ramirez, R. G., Ching, C. & Stlouis, R. D. 1993, 'Independence and mappings in model-based decision support systems', *Decision Support Systems,* vol. 10, no. 3, pp. 341-358.

Rao, H. R. & Chaudhury, A. 1995, 'Modeling team processes: Issues and a specific example', *Information Systems Research: ISR: A Journal of the Institute of Management Sciences,* vol. 6, no. 3, pp. 255-285.

Rao, H. R. & Lingaraj, B. P. 1988, 'Expert systems in production and operations management: Classification and prospects', *Interfaces,* vol. 18, no. 6, pp. 80-91.

Rao, H. R., Sridhar, R. & Narain, S. 1994, 'An active intelligent decision support system - architecture and simulation', *Decision Support Systems,* vol. 12, no. 1, pp. 79-91.

Rathnam, S. & Mannino, M. V. 1995, 'Tools for building the human-computer interface of a decision support system', *Decision Support Systems,* vol. 13, no. 1, pp. 35-59.

Rathwell, M. A. & Burns, A. 1985, ' Information systems support for group planning and decision-making activities', *MIS Quarterly,* vol. 9, no. 3, pp. 255-272.

Reisman, S., Johnson, T. W. & Mayes, B. T. 1992, 'Group decision program: A videodisc-based group decision support system', *Decision Support Systems,* vol. 8, no. 2, pp. 169-180.

Reitman, W. 1990, 'Generic expert systems for management applications: The operations advisor and the management advisor', *Comp. Sci. Econ. Mange.,* vol. 3, no. 2, pp. 167-175.

Remus, W. E. & Kottemann, J. E. 1986, 'Toward iIntelligent decision support systems: An artificially intelligent statistician', *MIS Quarterly,* vol. 10, no. 4, pp. 403-418.

Ribeiro, R. A. 1996, 'Fuzzy multiple attribute decision making - a review and new preference elicitation techniques', *Fuzzy Sets & Systems,* vol. 78, no. 2, pp. 155-181.

Rietveld, P. 1980, *Multiple objective decision methods and regional planning,* North-Holland, Amsterdam.

Ristroph, J. H., Muralidharan, R., Miglani, N., Arcuri, K. B. & Knight, M. 1993, 'Pollution prevention decision support system', *Computers & Industrial Engineering,* vol. 25, no. 1-4, pp. 349-352.

Roy, B. 1973, 'How outranking helps multiple criteria decision making', in *Multiple Criteria Decision Making,* ed. J. L. C. a. M. Zeleny, University of South Carolina Press, .

Rudowski, R., East, T. D. & Gardner, R. M. 1996, 'Current status of mechanical ventilation decision support systems - a review', *International Journal of Clinical Monitoring & Computing,* vol. 13, no. 3, pp. 157-166.

Sage, A. P. 1991, *Decision support systems engineering,* Weley.

Salminen, P., Korhonen, P. & Wallenius, J. 1989, 'Testing the form of a decision-maker's multiattribute value function based on pairwise preference information', *Journal of the Operational Research Society,* vol. 40, no. 3, pp. 299-302.

Salminen, P., Teich, J. E. & Wallenius, J. 1998, *The secretary problem revisited-the group decision-making perspective,* kluwer Academic Publishers, Netherland.

Sankar, C. S., Ford, F. N. & Bauer, M. 1995, 'A DSS user interface model to provide consistency and adaptability', *Decision Support Systems,* vol. 13, no. 1, pp. 93-104.

Saoly, T. L. 1980, *The analytic hierarchy process,* McGraw-Hill, New York.

Saxena, K. B. C. 1992, 'Dss development methodologies: A comparative review', in *The Twenty-fifth Annual Hawaii International Conference on System Sciences,* IEEE Computer Society Press, Los Alamitos.

Scanlan, J. C. & McKeon, G. M. 1993, 'Tropical pasture establishment .18. the potential for computer-based decision support systems to improve establishment success for sown tropical pastures', *Tropical Grasslands,* vol. 27, no. 4, pp. 414-419.

Schwartz, A. E. 1994, 'Group decision-making', *CPA Journal,* vol. 64, no. 8, pp. 60-63.

Schwartz, A. E. & Levin, J. 1990, 'Better group decision making', *Supervisory Management,* vol. 35, no. 6, p. 4.

Seilheimer, S. D. 1988, 'Current state of decision support system and expert system technology', *Journal of Systems Management,* vol. 39, no. 8, pp. 14-19.

Sen, A. & Biswas, G. 1985, 'Decision support systems: An expert systems approach', *Decision Support Systems,* vol. 1, no. 3, pp. 197-204.

Sen, P., Tan, J. K. G. & Spencer, D. 1999, 'An integrated probabilistic risk analysis decision support methodology for systems with multiple state variables', *Reliability Engineering & System Safety,* vol. 64, no. 1, pp. 73-87.

Sen, P. & Yang, J. B. 1994, 'Design decision making based upon multiple attribute evaluations and minimal preference information', *Mathematical & Computer Modelling,* vol. 20, no. 3, pp. 107-124.

Seng, Q. T., Lim, T. C., S, R. K. & Bobby, S. 1996, 'Towards integrating rule-based expert systems and neural networks', *Decision Support Systems,* vol. 17, no. 2, pp. 99-118.

Senn, J. A. 1990, *Information system in management,* Wadsworth Publishing Co., Belmont, California.

Sequeira, R. A., Willers, J. L. & Olson, R. L. 1996, 'Validation of a deterministic model-based decision support system', *AI Applications,* vol. 10, no. 1, pp. 25-40.

Shahsavar, N., Ludwigs, U., Blomqvist, H., Gill, H., Wigertz, O. & Matell, G. 1995, 'Evaluation of a knowledge-based decision support system for ventilator therapy management', *Artificial Intelligence in Medicine,* vol. 7, no. 1, pp. 37-52.

Shan, F. & Xu, L. D. 1996, 'A hybrid knowledge-based system for urban development', *Expert Systems with Applications,* vol. 10, no. 1, pp. 157-163.

Sharda, R. & Steiger, D. M. 1996, 'Inductive model analysis systems - enhancing model analysis in decision support systems', *Information Systems Research,* vol. 7, no. 3, pp. 328-341.

Shin, W. S. & Ravindran, A. 1991, ' Interactive multiple objective optimization: Survey I - continuous case', *Computers and Operations Research,* vol. 18, no. 1, pp. 97 - 114.

Shirani, A., Aiken, M. & Paolillo, J. 1998, 'Group decision support systems and incentive structures', *Information & Management,* vol. 33, no. 5, pp. 231-240.

Simon, H. 1960, *The new science of management decisions,* Harper & Row, New York.

Sinclair-Desgagne, B. 1990, 'Group decision support systems implement Bayesian rationality', *European Journal of Operational Research,* vol. 47, no. 1, pp. 29-34.

Singpurwalla, N. D. & Wilson, S. P. 1999, *Statistical methods in software engineering,* Springer - Verlag, New York.

Siskos, Y., Zopounidis, C. & Pouliezos, A. 1994, 'An integrated dss for financing firms by an industrial development bank in greece', *Decision Support Systems,* vol. 12, no. 2, pp. 151-168.

Snead, K. C. & Harrell, A. M. 1994, 'An application of expectancy theory to explain an managers intention tu use a decision support system', *Decision Sciences,* vol. 25, no. 4, pp. 499-513.

Sniezek, J. A. 1992, 'Groups under uncertainty: An examination of confidence in group decision making', *Organizational Behavior and Human Decision Processes,* vol. 52, no. 1, pp. 124-155.

Sol, H. G. 1985, 'DSS: Buzz word or OR challenge', *European Journal of Operational Research,* vol. 22, no. 1, pp. 1-8.

Sol, H. G. 1987, 'Conflicting experiences with DSS', *Decision Support Systems,* vol. 3, no. 3, pp. 203-211.

Sosik, J. J., Avolio, B. J. & Kahai, S. S. 1997, 'Effects of leadership style and anonymity on group potency and effectiveness in a group decision support system environment', *Journal of Applied Psychology,* vol. 82, no. 1, pp. 89-103.

Spangler, W. 1991, 'The role of artificial intelligence in understanding the strategic decision-making process', *IEEE Ttransaction on Knowledge and Data Engineering,* vol. 3, no. 2, pp. P145-159.

Sprague, R., H. & Hugh, J. W. 1995, *Decision support for management,* Prentice Hall, New Jersey.

Sprague, R. H. 1980, 'A framework for the development of decision support systems', *Management Information System Quarterly,* vol. 4, no. winter, pp. 1-26.

Sprague, R. H. 1987, 'DSS in context', *Decision Support Systems,* vol. 3, no. 3, pp. 197-202.

Sprague, R. H. & Watson, H. J. 1993, *Decision support systems: Putting theory into practice,* 3nd edition edn, Prentice-Hall, London.

Stephenson, R. W. & Stephenson, M. K. 1983, 'Design requirements for decision support systems for RDT&E', *Information Processing & Management,,* vol. 19, no. 6, pp. 391-398.

Steuer, R. E. 1977, 'An interactive multiple objective linear programming procedure', *TIMS Studies in the Management Sciences,* vol. 6, pp. 225-239.

Steuer, R. E. 1986, *Multiple criteria optimization: Theory, computation, and application,* Wiley, New York.

Steuer, R. E. & Choo, E. U. 1983, 'An interactive weighted tchebycheff procedure for multiple objective programming', *Mathematical Programming,* vol. 26, no. 1, pp. 326-344.

Stewart, T. J. 1991, 'A multi-criteria decision support system for R&D project selection', *Journal of the Operational Research Society,* vol. 42, no. 1, pp. 17-26.

Stewart, T. J. 1993, 'Use of piecewise linear value functions in interactive multicriteria decision support - a monte-carlo study', *Management Science,* vol. 39, no. 11, pp. 1369-1381.

Sturman, M. C., Hannon, J. M. & Milkovich, G. T. 1996, 'Computerized decision aids for flexible benefits decisions - the effects of an expert system and decision support system on employee intentions and satisfaction with benefits', *Personnel Psychology,* vol. 49, no. 4, pp. 883-908.

Sun, M. & Steuer, R. E. 1996, 'InterQuad: An interactive quad tree based procedure for solving the discrete alternative multiple criteria', *European Journal of Operational Research,* vol. 89, no. 3, pp. 462-472.

Sutherland, D. & Crosslin, R. 1989, 'Group decision support systems: Factors in a software implementation', *Information & Management,* vol. 16, no. 2, pp. 93-103.

Suzanne, P. & Pavlos, M. 1996, 'An intelligent distributed system for strategic decision making', *Group Decision and Negotiation,* vol. 6, pp. 77-108.

Sygenex 1989, *CRITERIUM: The complete decision making system,* Redmond, USA.

Tabucanon, M. T., Batanov, D. N. & Verma, D. K. 1994, 'Decision support system for multicriteria machine selection for flexible manufacturing systems', *Computers in Industry,* vol. 25, no. 2, pp. 131-143.

Taha, M. A., Park, S. C. & Russell, J. S. 1995, 'Knowledge-based dss for construction contractor prescreening', *European Journal of Operational Research,* vol. 84, no. 1, pp. 35-46.

Tecle, A. & Duckstein, L. (eds.) 1992, *A procedure for selecting MCDM techniques for forest resource management,* Springer-Verlag, New York.

Tecle, A., Shrestha, B. P. & Duckstein, L. 1998, 'A multiobjective decision support system for multiresource forest management', *Group Decision and Negotiation,* vol. 7, no. 1, pp. 23-40.

Te'eni, D. & Ginzberg, M. J. 1991, 'Human-computer decision systems: The multiple roles of DSS', *European Journal of Operational Research,* vol. 50, no. 2, pp. 127-139.

Teng, J. T. & Galetta, D. F. 1990, 'MIS research directions: A survey of researcher's views', *Data Base,* vol. 21, no. fall, pp. 1-10.

Teng, J. T. C., Mirani, R. & Sinha, A. 1988, 'A unified architecture for intelliogent DSS', in *The 21st Hawaii International Conference on System Sciences,* pp. 286-294.

Thierauf, R. J. 1982, *Decision support systems for effective planning and control,* Prentice-Hall, Inc., Englewood Cliffs.

Travica, B. & Cronin, B. 1995, 'The argo: A strategic information system for group decision making', *International Journal of Information Management,* vol. 15, no. 3, pp. 223-236.

Tsoukias, A. 1991, 'Preference modeling as a reasoning process: A new way to face uncertainty in multiple criteria decision support systems', *European Journal of Operational Research,* vol. 55, pp. 309-318.

Tung, L. L. & Turban, E. 1998, 'A proposed research framework for distributed group support systems', *Decision Support Systems,* vol. 23, no. 2, pp. 175-188.

Turban, E. 1995, *Decision support and expert systems: Management support systems,* 4 edn, Prentice-Hall, Inc., New Jersey.

Turban, E. 1996, 'Implementing decision support system: a survey', in *The 1996 IEEE International Conference on Systems, Man and Cybernetics,* IEEE, PISCATAWAY (USA), Beijing, china.

Turban, E. & Erikson, W. 1985, 'Selecting operations research software for classroom microcomputers', *Computers and Operations Research,* vol. 12, no. 4, pp. 383-390.

Turban, E. & Sepehri, M. 1986, 'Applications of decision support and expert systems in flexible manufacturing systems', *Journal of Operations Management,* vol. 6, no. 3/4, pp. 433-448.

Turban, E. & Watkins, P. R. 1986, 'Integrating expert systems and decision support systems', *MIS Quarterly,* vol. 10, no. 2, pp. 121-136.

Tzeng, G. H. & Chen, C. H. 1993, 'Multiobjective decision making for traffic assignment', *IEEE Transactions on Engineering Management,* vol. 40, no. 2, pp. 180-187.

Tzeng, G. H., Shiau, T. A. & Teng, J. Y. 1994, 'Multiobjective decision-making approach to energy supply mix decisions in taiwan', *Energy Sources,* vol. 16, no. 3, pp. 301-316.

Valacich, J. S. & Schwenk, C. 1995, 'Structuring conflict in individual, face-to-face, and computer-mediated group decision making: Carping versus objective devil's advocacy', *Decision Sciences,* vol. 26, no. 3, pp. 369-393.

Van, H. K. M. & Lapinski, A. 1988, 'OR and AI approaches to decision support systems', *Decision Support Systems,* vol. 4, no. 4, pp. 447-459.

Vanhee, K. M. & Lenstra, J. K. 1994, 'A comparative study in DSS development', *European Journal of Operational Research,* vol. 79, no. 2, pp. 153-157.

Venkatachalam, A. R. & Sohl, J. E. 1999, ' An intelligent model selection and forecasting system', *Journal of Forecasting,* vol. 18, no. 3, pp. 167-180.

Vetschera & Rudolf 1994, 'MCView: An integrated graphical system to support multi-attribute decision', *Decision Support Systems,* vol. 11, no. 4, pp. 363-371.

Vlacic, W., Amagasa, M., Ishikawa, A., Williams, T. & Tomizawa, G. 1997, 'Applying multisttribute-based group decision making techniques in complex equipment selection tasks', *Group Decision and Negotiation,* vol. 6, no. 6, pp. 529-556.

Vogel, D. R. & Nunamaker, J. F., Jr. 1990, 'Group decision support system impact: Multi-methodological exploration', *Information and Management,* vol. 18, no. 1, pp. 15-28.

Volkema, R. J. & Niederman, F. 1995, 'Meeting the challenge: Application of communication technologies to group interactions', in *The 1995 ACM Sigcpr Conference,* ACM, New York, NY (USA), New York, pp. 131-138.

Wang, H. 1997a, 'Intelligent agent-assisted decision support systems: Interaction of knowledge discovery, knowledge analysis, and group decision support', *Expert Systems with Applications,* vol. 12, no. 3, pp. 323-335.

Wang, H. 1997b, 'knowledge level analysis of group decision support systems', *Expert Systems with Applications,* vol. 13, no. 2, pp. 155-162.

Wang, J., Yang, J. B., Sen, P. & Ruxton, T. 1996, 'Safety based design and maintenance optimisation of large marine engineering systems', *Applied Ocean Research,* vol. 18, no. 1, pp. 13-27.

Wang, P. Z. 1985, 'A fuzzy evaluation approach', *Journal of Beijing Teachers University,* vol. 2, no. 8, pp. 21-35 (in Chinese).

Wang, Z. M. 1995, 'Task constraints and user-system interaction process under personnel decision support', *Ergonomics,* vol. 38, no. 5, pp. 1049-1056.

Ward, J. C. & Reingen, P. H. 1990, 'Sociocognitive analysis of group decision making among consumers', *Journal of Consumer Research,* vol. 17, no. 3, pp. 245-262.

Watson, R. T., DeSanctis, G. & Poole, M. S. 1988, 'Using a GDSS to facilitate group consensus: Some intended and unintended consequences', *MIS quarterly,* vol. 12, no. 3, pp. 483-478.

Watson, S. R. & Buede, D. M. 1987, *Decision synthesis: The principles and practice of decision analysis,* Cambridge University Press, Cambridge.

Weiss, H. J. & Assous, J. Y. 1987, 'Reduction in problem size for ranking alternatives in group decision-making', *Computers and Operations Research,* vol. 14, no. 1, pp. 55-65.

Werners, B. 1987, 'Interactive multiple objective programming subject to flexible constraints', *European Journal of Operational Research,* vol. 31, no. 3, pp. 242-349.

White, C. C. 1990, 'A survey on the intelligent of decision analysis and expert systems for decision support', *IEEE: Transactions on Systems Man and Cybernetics,* vol. 20, no. 2, pp. 358-364.

Wierenga, B., Ophuis, P. A. M. O., Huizingh, E. K. R. & Vancampen, P. A. F. M. 1994, 'Hierarchical scaling of marketing decision support systems', *Decision Support Systems,* vol. 12, no. 3, pp. 219-232.

Wierzchon, S. T. 1983, ' An algorithm for identication of fuzzy measure', *Fuzzy Sets and Systems,* vol. 9, no. 1, pp. 69-78.

Wigren, T. & Kolari, P. 1994, 'Evaluation of a decision-support system for inoperable non-small cell lung cancer', *Methods of Information in Medicine,* vol. 33, no. 4, pp. 397-401.

Wilkes, G. A. & Krebs, W. A. 1995, *Collins English dictionary: An extensive coverage of contemporary international and Australia English,* HarperCollins Publishers, Sydney.

Willett, M. J. & Andrews, P. K. 1996, 'Knowledge-based systems for fruit and vegetable production management', *AI Applications,* vol. 10, no. 1, pp. 75-92.

Williams, R., Klass, D. & Morien, R. 1997, 'Developing the ALLOCATE resource allocation system: A case study in prototyping a decision support system', in *The 8th Australasian Conference on Information Systems,* 745-754, University of South Australia, Adelaide, South Australia,.

Wolf, G. 1994, 'Schedule management - an object oriented approach', *Decision Support Systems,* vol. 11, no. 4, pp. 373-388.

Wolfe, M. 1994, 'Development of the city of quality - a hypertext-based group decision support system for quality function deployment', *Decision Support Systems,* vol. 11, no. 3, pp. 299-318.

Wolfe, M. D. 1992, 'A Theoretical justification for Japanese nemawashi/ringi group decision making and an implementation of a nemawashi/ringi group decision support system', *Decision Support Systems,* vol. 8, no. 2, pp. 125-140.

Xia, H., Iz, P. H. & Shi, X. 1996, 'Bi-level decision and negotiation based on multicriteria optimization and game theory', in *The 1996 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, PISCATAWAY (USA), Beijing, China.

Xu, L. D. 1994, 'A decision support system for aids intervention and prevention', *International Journal of Bio Medical Computing,* vol. 36, no. 4, pp. 281-291.

Yakowitz, D. S. & Hipel, K. W. 1997, 'Multiple objective decision making for lokahi (balance) in environmental management', *Applied Mathematics & Computation,* vol. 83, no. 2-3, pp. 97-115.

Yakowitz, D. S., Stone, J. J., Lane, L. J., Heilman, P., Masterson, J., Abolt, J. & Imam, B. 1993, 'A decision support system for evaluating the effects of alternative farm management systems on water quality and economics', *Water Science & Technology,* vol. 28, no. 3-5, pp. 47-54.

Yang, H. L. 1995, 'Information knowledge acquisition methods for decision support systems and expert systems', *Information Processing & Management,* vol. 31, no. 1, pp. 47-58.

Yau, C. & Davis, T. 1994, 'Using multi-criteria analysis for tenant selection', *Decision Support Systems,* vol. 12, no. 3, pp. 233-244.

Yellen, R. 1993, 'Introducing group decision support software (GDSS) in an organization', *Journal of System Management,* vol. 44, no. 10, pp. 6-8.

Zannetos, Z. S. 1968, 'Toward intelligent managent information systems', *Industrial Management Review,* vol. 9, no. 3, pp. 21-28.

Zeleny, M. 1982, *Multiple criteria decision making,* McGraw-Hill, New York.

Zhuo, Y., Wu, Q. & Wang, H. 1996, 'Distributed group decision support system with intelligent multimedia', in *The 1996 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, PISCATAWAY, NJ, (USA), Beijing, China, pp. 2070-2073.

Zigurs, I., Poole, M. S. & DeSanctis, G. L. 1988, 'A study of influence in computer-mediated group decision making', *MIS Quarterly,* vol. 12, no. 4, pp. 625-644.

Zionts, S. & Wallenius, J. 1976, 'An interactive programming method for solving the multiple criteria problem', *Management Science,* vol. 22, no. 6, pp. 652-663.

Zwibelman, B. B. & Plant, R. T. 1994, 'Choosing a college major - a prototype decision support system', *Computers in Human Behavior,* vol. 10, no. 3, pp. 231-242.

Appendix 1 Main Program of DELPHI in IMOGDSS

```
Program PImogdss;
uses
  Forms,
  Main in 'Main.pas' {frmMain},
  ObjectInput in 'ObjectInput.pas' {frmObjectInput},
  ConstInput in 'ConstInput.pas' {frmConstraintInput},
  CommonVariables in 'CommonVariables.pas',
  UserModelCurrent in 'UserModelCurrent.pas' {frmUserModelCurrent},
  GuideNovice1 in 'GuideNovice1.pas' {frmGuideNovice1},
  GuideIntermediate in 'GuideIntermediate.pas' {frmGuideIntermediate},
  DecVarInput in 'DecVarInput.pas' {frmDecVarInput},
  ObjVarInput in 'ObjVarInput.pas' {frmObjVarInput},
  ConVarInput in 'ConVarInput.pas' {frmConVarInput},
  VarConvert in 'VarConvert.pas',
  ResultCurrent in 'ResultCurrent.pas' {frmresultCurrent},
  LGP1 in 'LGP1.pas' {frmLGP1},
  LGP2 in 'LGP2.pas' {frmLGP2},
  Resultcurrent2 in 'Resultcurrent2.pas' {frmResultcurrent2},
  STEUER1 in 'STEUER1.pas' {frmSTEUER1},
  usermodelDB in 'usermodelDB.pas' {frmUserModelDB},
  ESGP in 'ESGP.pas' {frmESGP},
  STEM in 'STEM.pas' {frmSTEM},
  ISGP1 in 'ISGP1.pas' {frmISGP},
  IMOLP in 'IMOLP.pas' {frmIMOLP},
  LP1 in 'LP1.pas' {frmLP1},
  ResultDB in 'ResultDB.pas' {frmresultDB},
  ZW in 'ZW.pas' {frmZw},
  Report in 'Report.pas' {frmReport},
  CLIPSHeader in 'CLIPSHeader.pas',
  CLIPS in 'CLIPS.pas',
  CLIPSSupport in 'CLIPSSupport.pas',
  CLIPSFact in 'ClipsFact.pas',
  Knowledgebase in 'Knowledgebase.pas' {frmKnowledgebase},
  OpenSolutionFile in 'OpenSolutionFile.pas' {FrmOpenSolution},
  AggregationASM in 'aggregationASM.pas' {frmASM},
  Averagechart in 'averagechart.pas' {frmAverageChart},
  Inputsolution in 'inputsolution.pas' {frminputSolution},
  AggregationWOM in 'AggregationWOM.pas' {frmWOM},
  InputWeightGroup in 'inputWeightGroup.pas' {frmWeight},
  GroupCommonVar in 'GroupCommonVar.pas',
  Inputsolution1 in 'inputsolution1.pas' {frminputsolution1},
  ResultChart1 in 'resultChart1.pas' {frmresultChart},
```

```
  GroupReport in 'groupReport.pas' {frmGroupReport},
  AggregationWMM in 'AggregationWMM.pas' {frmWMM},
  AggregationISM in 'AggregationISM.pas' {frmISM},
  AggregationSAM in 'AggregationSAM.pas' {frmSAM},
  IdealChart in 'IdealChart.pas' {frmIdealChart},
  GroupAnalyseUsingSTEM in 'GroupAnalyseUsingSTEM.pas' {frmUsingSTEM};

{$R *.RES}

begin
  Application.Initialize;

  Application.Title := 'IMOGDSS';
  Application.CreateForm(TfrmMain, frmMain);
  Application.CreateForm(TfrmObjectInput, frmObjectInput);
  Application.CreateForm(TfrmConstraintInput, frmConstraintInput);
  Application.CreateForm(TfrmUserModelCurrent, frmUserModelCurrent);
  Application.CreateForm(TfrmGuideNovice1, frmGuideNovice1);
  Application.CreateForm(TfrmGuideIntermediate, frmGuideIntermediate);
  Application.CreateForm(TfrmDecVarInput, frmDecVarInput);
  Application.CreateForm(TfrmObjVarInput, frmObjVarInput);
  Application.CreateForm(TfrmConVarInput, frmConVarInput);
  Application.CreateForm(TfrmresultCurrent, frmresultCurrent);
  Application.CreateForm(TfrmLGP1, frmLGP1);
  Application.CreateForm(TfrmLGP2, frmLGP2);
  Application.CreateForm(TfrmResultcurrent2, frmResultcurrent2);
  Application.CreateForm(TfrmSTEUER1, frmSTEUER1);
  Application.CreateForm(TfrmUserModelDB, frmUserModelDB);
  Application.CreateForm(TfrmESGP, frmESGP);
  Application.CreateForm(TfrmSTEM, frmSTEM);
  Application.CreateForm(TfrmISGP, frmISGP);
  Application.CreateForm(TfrmIMOLP, frmIMOLP);
  Application.CreateForm(TfrmLP1, frmLP1);
  Application.CreateForm(TfrmresultDB, frmresultDB);
  Application.CreateForm(TfrmZw, frmZw);
  Application.CreateForm(TfrmsolutionReport, frmsolutionReport);
  Application.CreateForm(TfrmKnowledgebase, frmKnowledgebase);
  Application.CreateForm(TFrmOpenSolution, FrmOpenSolution);
  Application.CreateForm(TfrmASM, frmASM);
  Application.CreateForm(TfrmAverageChart, frmAverageChart);
  Application.CreateForm(TfrminputSolution, frminputSolution);
  Application.CreateForm(TfrmWOM, frmWOM);
  Application.CreateForm(TfrmWeight, frmWeight);
  Application.CreateForm(Tfrminputsolution1, frminputsolution1);
  Application.CreateForm(TfrmresultChart, frmresultChart);
  Application.CreateForm(TfrmGroupReport, frmGroupReport);
```

```
  Application.CreateForm(TfrmWMM, frmWMM);
  Application.CreateForm(TfrmISM, frmISM);
  Application.CreateForm(TfrmSAM, frmSAM);
  Application.CreateForm(TfrmIdealChart, frmIdealChart);
  Application.CreateForm(TfrmUsingSTEM, frmUsingSTEM);
  Application.Run;
end.
```

Note:

The full program consists of over 40,000 lines of codes.

Appendix 2 Evaluation Form

Group Decision-Maker Evaluation (GDME) of IMOGDSS

Name:________________________________ **Sex (M/F):**________ **Date:**______________

Your study background:
business/management, information systems & technology, mathematics, engineering, others
Your work background:
business/management, information systems & technology, mathematics, engineering, others

Instructions for using GDME

1. The GDME will help the developer of IMOGDSS to test and evaluate the system and thus improve it further. Your feedback will be invaluable to this research.
2. Please don't put your name on the form if you don't want to but do fill in the details asked for at the top of the form.
3. After you have filled in the appropriate sets of questions please complete the open-ended comment sections on the back of the form.
4. Complete the form as accurately and honestly as you can.

Please mark the items, if any items are not applicable simply leave them blank

Question (Users-oriented)	***Items***
Which **mode** of intelligent guide have you used?	Novice, Intermediate, Expert
Which MODM **method**(s) have you used?	ESGP, IMOLP, ISGP, LGP, STEM, STEUER, ZW
How long have you taken to solving an application?	20min or less, 30min, 40min, 50min, 60mim or more
Do you have the background **knowledge** for	Multiple objective decision making (MODM) or multiple criteria decision making (MADM)
Do you have the background **knowledge** for	Decision support systems (DSS), Artificial intelligence (AI) & Expert systems (ES)
Do you have the ability to **select an** appropriate MODM method for application without the expert system based guidance?	Yes, No
Do you like to **interact** with system during the solution process?	Strongly like, like, don't like

Please indicate the EXTENT of your agreement/disagreement and the DEGREE OF IMPORTANCE (WEIGHTS) of each question with the following statements on IMOGDSS

	1	*2*	*3*	*4*	*5*
Extent	*strongly disagree*	*disagree*	*some what agree*	*agree*	*strongly agree*
Weight	*not important*	*less important*	*some what important*	*important*	*very important*

Questions (Input-oriented)	**Extent**	***Degree of importance***
The **organisation of input** in the system is good	1 2 3 4 5	1 2 3 4 5
The input approaches in the system are **easy** to understand	1 2 3 4 5	1 2 3 4 5
It is **easy to change** the input value	1 2 3 4 5	1 2 3 4 5
Questions (Interactive Solution Process-oriented)	***Extent***	***Degree of importance***
You found the feedback **messages** on screen **valuable**	1 2 3 4 5	1 2 3 4 5
The solution process is **friendly** towards a decision maker	1 2 3 4 5	1 2 3 4 5
This system solution process is ease to **understand**	1 2 3 4 5	1 2 3 4 5
It is very **easy to interact** with the system for finding a satisfactory solution	1 2 3 4 5	1 2 3 4 5
You are satisfied with the solution **interactive process**	1 2 3 4 5	1 2 3 4 5
It is easy to make **"what-if" analysis**	1 2 3 4 5	1 2 3 4 5
Questions (Output-oriented)	***Extent***	***Degree of importance***
The **organisation** of output of this system is good	1 2 3 4 5	1 2 3 4 5
The **completeness** of the output of this system is good	1 2 3 4 5	1 2 3 4 5
The output of this system is **reliable**	1 2 3 4 5	1 2 3 4 5
The output of this system is **relevant**	1 2 3 4 5	1 2 3 4 5
Questions (Intelligent Guide-oriented)	***Extent***	***Degree of importance***
You think the **intelligent guide** is necessary	1 2 3 4 5	1 2 3 4 5
You found this intelligent subsystem really **useful and helpful** for a suitable MODM method selection	1 2 3 4 5	1 2 3 4 5
You think the three intelligent **guide modes** are suitable	1 2 3 4 5	1 2 3 4 5
You think this intelligent guide subsystem is **easy to use**	1 2 3 4 5	1 2 3 4 5
This intelligent guide system really enhances the **quality** of decision-making	1 2 3 4 5	1 2 3 4 5
Questions (Overall Ability and system performance-oriented)	***Extent***	***Degree of importance***
You found this system is **efficient** in task performance (decision making time)	1 2 3 4 5	1 2 3 4 5
This system really enhances the **quality** of decision-making	1 2 3 4 5	1 2 3 4 5
This system is **flexible** where users are not constrained and have the option to explore and direct the solution process as their wish	1 2 3 4 5	1 2 3 4 5
This system uses **graphical user interface** very well	1 2 3 4 5	1 2 3 4 5
Do you think this system is **effective?**	1 2 3 4 5	1 2 3 4 5
Do you think this system is **easy to use?**	1 2 3 4 5	1 2 3 4 5
Are you **satisfied** with the ability of MODM methods towards a satisfactory solution	1 2 3 4 5	1 2 3 4 5
Do you feel **confident** in the final solution as being a good representative of your preferences?	1 2 3 4 5	1 2 3 4 5
This system can solve some **complex** decision problems	1 2 3 4 5	1 2 3 4 5

Open-ended comment

____

____

____

Group Decision-Maker Evaluation (GDME) of IMOGDSS

(For Group Subsystem only)

Name:________________________ **Sex (M/F):**______ **Date:**__________

Your study background:
business/management, information systems & technology, mathematics, engineering, others
Your work background:
business/management, information systems & technology, mathematics, engineering, others

Question (Users-oriented)	***Items***
Which **managerial role** (function area of the company) did you taken in your group?	Finance, marking, operations management, quality controller, general manager, others
Which **form** of group meeting have you taken?	Face-to-face (FTF), same time & different places, same place & different time, different time & places
Which style of group meeting have you taken?	Group meeting with a leader and different weights for different members, same weights for all members and no-leader
Which group aggregation **method**(s) have you uses?	ASM, WOM, WMM, ISM, SAM
How long have you taken to reach a compromise solution of the application?	20min or less, 30min, 40min, 50min, 60mim or more
Do you have the background **knowledge** for	Group decision support systems (GDSS) or Group support systems (GSS)

Please indicate the EXTENT of your agreement/disagreement and the DEGREE OF IMPORTANCE (WEIGHTS) of each question with the following statements on IMOGDSS

	1	2	3	4	5
Extent	*strongly disagree*	*disagree*	*some what agree*	*agree*	*strongly agree*
Weight	*not important*	*less important*	*some what important*	*important*	*very important*

Questions (Input-oriented)	***Extent***	***Degree of importance***
The organisation of input in the group subsystem is good	1 2 3 4 5	1 2 3 4 5
The input approaches in the group subsystem are easy	1 2 3 4 5	1 2 3 4 5
It is ease to change the input value	1 2 3 4 5	1 2 3 4 5
Questions (Interactive Process-oriented)	***Extent***	***Degree of importance***
You found the feedback on screen messages valuable	1 2 3 4 5	1 2 3 4 5
The group solution process is friendly towards all group members	1 2 3 4 5	1 2 3 4 5

It is very easy to interact with the group subsystem for finding a compromise solution	1 2 3 4 5	1 2 3 4 5
You are satisfied with the interactive process of this subsystem	1 2 3 4 5	1 2 3 4 5
Questions (Output-oriented)	***Extent***	***Degree of importance***
The organisation of output of this subsystem is good	1 2 3 4 5	1 2 3 4 5
The completeness of the output of this group subsystem is good	1 2 3 4 5	1 2 3 4 5
The output of this group subsystem is reliable	1 2 3 4 5	1 2 3 4 5
Questions (Overall ability-oriented)	***Extent***	***Degree of importance***
The group aggregation method(s) you used is reliable to gain a compromise solution?	1 2 3 4 5	1 2 3 4 5
This group aggregation method(s) really enhances the quality of solution aggregation.	1 2 3 4 5	1 2 3 4 5
You think this group aggregation methodology base and the group subsystem is easy to use	1 2 3 4 5	1 2 3 4 5
Are you satisfied with this interactive process for finding a compromise solution?	1 2 3 4 5	1 2 3 4 5
The process for finding a compromise solution is friendly	1 2 3 4 5	1 2 3 4 5
Questions (System performance)	***Extent***	***Degree of importance***
The system is **efficient** for task performance (group aggregation time)	1 2 3 4 5	1 2 3 4 5
This group subsystem really enhances the **quality** of group decision-making	1 2 3 4 5	1 2 3 4 5
This system is **flexible** where users have the option to explore and direct the solution aggregation process as their wish	1 2 3 4 5	1 2 3 4 5
This system uses **graphical user interface** very well	1 2 3 4 5	1 2 3 4 5
Do you think this system is **effective?**	1 2 3 4 5	1 2 3 4 5
Are you **satisfied** with the final compromise solution?	1 2 3 4 5	1 2 3 4 5
Are you **satisfied** in the ability of these GDM methods towards a compromise solution?		
Do you feel **confident** in the group consensus solution as being a good representative of your preferences?	1 2 3 4 5	1 2 3 4 5
You have **learned** something from this group subsystem which you considered valuable	1 2 3 4 5	1 2 3 4 5
Are you satisfied with this **facilitator's work?**	1 2 3 4 5	1 2 3 4 5

Open-ended comment

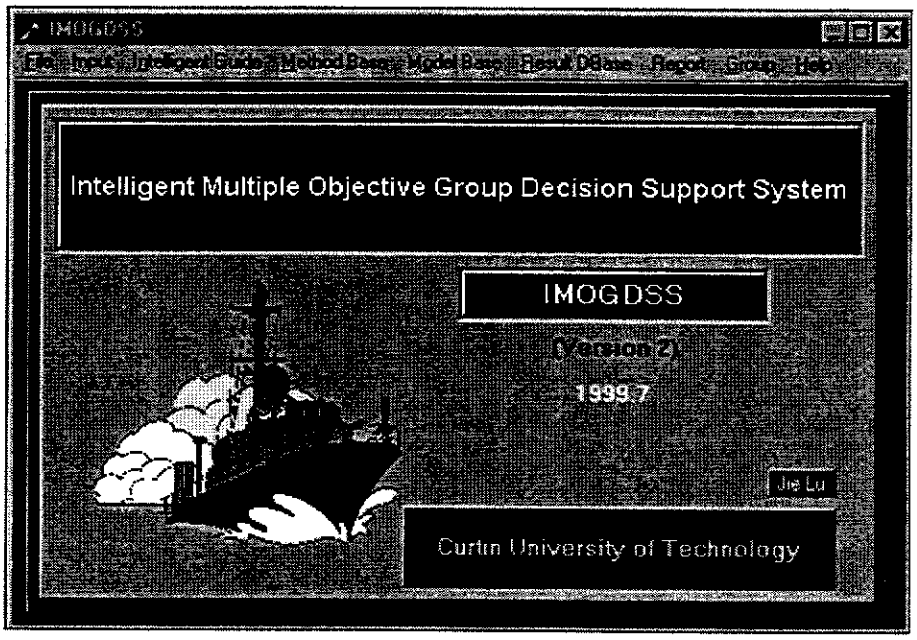
Thank you for your feedback

Appendix 3 User Manual (cover page and table of contents)

Getting Started With

Intelligent Multiple Objectives Group Decision Support System

(IMOGDSS)



Jie Lu

Contents

Glossary

Artificial Intelligence (AI)

Average Solution Method (ASM)

Computer Mediated Communication (CMC)

Data Base Management System (DBMS)

Decision Maker (DM)

Decision Support Systems (DSS)

Efficient Solution via Goal Programming (ESGP)

Expert Systems (ES)

Face to Face (FTF)

Group Decision Making (GDM)

Group Decision Support Systems (GDSS)

GDSS involving MCDM (MCGDSS)

GDSS involving MADM (MAGDSS)

GDSS involving MODM (MOGDSS)

Ideal Solution Method (ISM)

Intelligent DSS (IDSS)

Intelligent Group DSS (IGDSS)

Intelligent MCDSS (IMCDSS)

Intelligent MODSS (IMODSS)

Intelligent MOGDSS (IMOGDSS)

Interactive Multiple Objective Linear Program (IMOLP)

Interactive Sequential Goal Programming (ISGP)

Knowledge-Based DSS (KBDSS)

Linear Programming (LP)

Linear Goal Programming (LGP)

Multiple Criteria Decision Making (MCDM)

Multiple Attribute Decision Making (MADM)

Multiple-Objective Decision Making (MODM)

MADM model-based DSS (MADSS)

MODM model-based DSS (MODSS)

Model Base Management System (MBMS)

Step Method (STEM)

STEUER Method (STEUER)

Solution Analysis Method (SAM)

Weighting Objective Method (WOM)

Weighting Menber Method (WMM)

Zionts and Wallenius Method (ZW)